Skip lists: a randomized dictionary

Michiel Smid*

March 5, 2004

1 Introduction

We consider the *dictionary problem*: Given a set S of n real numbers, store them in a data structure such that the following three operations can be performed efficiently:

1. *Search*(x): Given the real number x, find the largest element of $S \cup \{-\infty\}$ that is less than or equal to x. (By introducing $-\infty$, this operation is always well-defined.)

2. *Insert*(x): Given the real number x, insert it into the dictionary. (Hence, we set $S := S \cup \{x\}$.)

3. *Delete*(x): Given the real number x, delete it from the dictionary. (Hence, we set $S := S \setminus \{x\}$.)

The standard data structure for this problem is a balanced binary search tree. It solves the dictionary problem in $O(\log n)$ worst-case time for each of the three operations, and it uses $O(n)$ space. Well known classes of balanced binary search trees are AVL-trees, BB[α]-trees and red-black-trees. Since these trees solve the dictionary problem optimally, we may think that the story ends here. However, anyone who has *implemented* a specific class of balanced trees will have noticed that especially the update algorithms are not trivial to code. Usually, the tree is rebalanced by means of rotations and

*School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6. E-mail: `michiel@scs.carleton.ca`.

double rotations. Moreover, each of these two types of rotations has a "left" and a "right" version. That is, we have to distinguish between different cases and, therefore, have to be extremely careful.

This leads to the question whether there is an optimal data structure for the dictionary problem that is easy to implement and that is also fast in practice. In this chapter, we will introduce such a data structure: the *skip list*. This data structure was invented by William Pugh in 1989. Skip lists use *randomization*, i.e., they use the outcomes of random coin flips. As we will see, we (i.e., the programmer) do *not* have to worry about balancing: the coin flips take care that the data structure is balanced, at least with a very high probability.

The rest of this chapter is organized as follows. We first define skip lists and give the algorithms that work on them. Then we give the intuition why skip lists are efficient. Next, we recall some basic notions from probability theory. Finally, we give complete proofs of the complexity of a skip list.

2 Skip lists

Throughout this chapter, we assume that we can generate random independent bits. Each bit can be generated in unit time. Put differently, we have a fair coin. If we flip it, then we obtain a one (heads) with probability 1/2 and a zero (tails) with probability 1/2. The outcome of a coin flip is independent of previous outcomes. By flipping the coin repeatedly, we obtain a sequence of independent random bits.

Let S be a set of n real numbers. We construct a sequence $S_1, S_2, S_3, \ldots$ of sets, in the following way.

1. $S_1 := S$.

2. Let $i \geq 1$, and assume that S_i has been constructed already. Flip the coin independently for each element of S_i. Then S_{i+1} is the set of all elements of S_i for which the coin flip produced a one.

3. The construction stops as soon as the current set S_i is empty.

Let h be the number of sets that are constructed. Then

$$\emptyset = S_h \subseteq S_{h-1} \subseteq \ldots \subseteq S_2 \subseteq S_1 = S.$$

The *skip list* for S consists of the following:

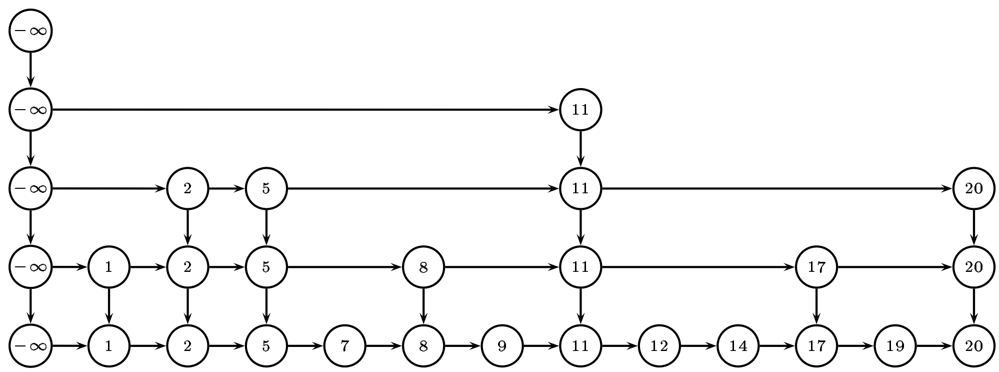

Figure 1: *A skip list.*

1. For each i, $1 \le i \le h$, the elements of $S_i \cup \{-\infty\}$ are stored in sorted order in a doubly-linked list L_i.

2. For each i, $1 < i \le h$, and each element x in L_i, there is a pointer from x to its occurrence in L_{i-1}.

Here is an example. Suppose $S = \{1, 2, 5, 7, 8, 9, 11, 12, 14, 17, 19, 20\}$. Flipping coins may lead to the sets $S_1 = S$, $S_2 = \{1, 2, 5, 8, 11, 17, 20\}$, $S_3 = \{2, 5, 11, 20\}$, $S_4 = \{11\}$, and $S_5 = \emptyset$. The corresponding skip list is shown in Figure 1.

Observation 2.1 *The construction of the sets S_i, $i \ge 1$, defines a probability distribution on skip lists. In our example, the probability that the skip list of Figure 1 is obtained is exactly the probability that the coin flipping process gives the sets $S_2 = \{1, 2, 5, 8, 11, 17, 20\}$, $S_3 = \{2, 5, 11, 20\}$, $S_4 = \{11\}$, and $S_5 = \emptyset$.*

We can search in a skip list as follows. Let $x \in \mathbb{R}$. Recall that we want to find the largest element of $S \cup \{-\infty\}$ that is less than or equal to x. The algorithm successively locates x in the lists $L_h, L_{h-1}, \dots, L_2, L_1$:

1. Let y_h be the only element of L_h.

2. For $i = h, h-1, \dots, 2$:

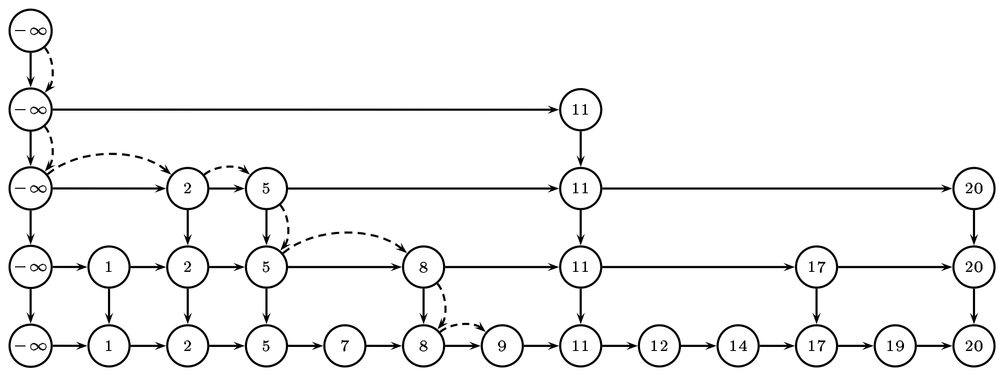

Figure 2: *Searching for 9.*

(a) Follow the pointer from y_i in L_i to its occurrence in L_{i-1}.

(b) Starting in y_i, walk to the right in L_{i-1}, until an element is encountered that is larger than x, or the end of L_{i-1} is reached. Let y_{i-1} be the last element in L_{i-1} encountered that is less than or equal to x.

3. Return y_1.

See Figure 2 for an example, in which we search for the number 9. The search path consists of the dashed arrows. The y-variables have values $y_5 = -\infty$, $y_4 = -\infty$, $y_3 = 5$, $y_2 = 8$, and $y_1 = 9$.

Exercise 2.2 Convince yourself that the search algorithm is correct.

Before we turn to the insertion and deletion algorithms, we give an *alternative* construction of the sets S_i:

1. For each element x of S, flip the coin until a zero comes up.

2. For each $i \geq 1$, S_i is the set of all elements in S for which we flipped the coin at least i times.

Observe that the sets S_i, $i \geq 1$, completely determine the skip list. The alternative construction also defines a probability distribution on skip lists.

Exercise 2.3 Convince yourself that the two given constructions of the sets S_i define the *same* probability distribution on skip lists. (Hint: In both constructions, the coin flips are independent.)

The alternative construction immediately suggests the following insertion algorithm. Let $x \in \mathbb{R}$ be the element to be inserted.

1. Run the search algorithm on x. Let $y_h, y_{h-1}, \ldots, y_1$ be the elements of $L_h, L_{h-1}, \ldots, L_1$, respectively, that are computed by this algorithm. If $x = y_1$, then $x \in S$ and nothing has to be done. So assume that $x \neq y_1$.

2. Flip the coin until a zero comes up. Let ℓ be the number of coin flips.

3. For each i, $1 \leq i \leq \min(\ell, h)$, add x to the list L_i, as the successor of y_i.

4. If $\ell \geq h$, then create new lists $L_{h+1}, L_{h+2}, \ldots, L_{\ell+1}$ storing the sets $S_{h+1} \cup \{-\infty\}$, $S_{h+2} \cup \{-\infty\}$, $\ldots$, $S_{\ell+1} \cup \{-\infty\}$, where $S_{h+1} = S_{h+2} = \ldots = S_\ell = \{x\}$ and $S_{\ell+1} = \emptyset$.

5. For each i, $1 < i \leq \ell$, give x in L_i a pointer to its occurrence in L_{i-1}.

6. If $\ell \geq h$, then for each i, $h+1 \leq i \leq \ell+1$, give $-\infty$ in L_i a pointer to its occurrence in L_{i-1}.

7. Set $h := \max(h, \ell+1)$.

In Figure 3, the skip list that arises by inserting 10 into the skip list of Figure 1 is depicted. The dashed pointers are the new ones.

The deletion algorithm is similar. Suppose we want to delete element x.

1. Run the search algorithm on x. Let $y_h, y_{h-1}, \ldots, y_1$ be the elements of $L_h, L_{h-1}, \ldots, L_1$, respectively, that are computed by this algorithm. If $x \neq y_1$, then $x \notin S$ and nothing has to be done. So assume that $x = y_1$.

2. For each i, $1 \leq i \leq h$ such that $x = y_i$, delete y_i from the list L_i.

3. For each $i = h$, $h-1$, $\ldots$: if L_{i-1} only stores $-\infty$, then delete the list L_i and set $h := h - 1$.

Exercise 2.4 Delete 11 from the skip list of Figure 3.

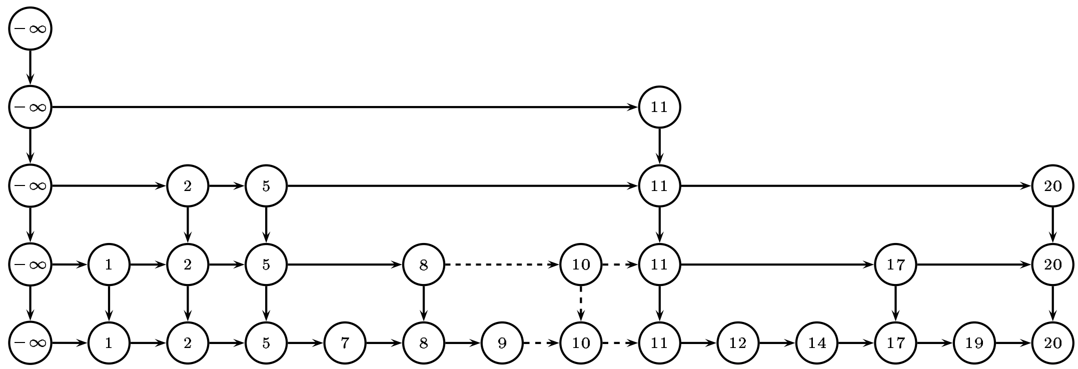

Figure 3: *Inserting 10. The first flip gives a one, the second a zero. Hence, 10 is added to the first two levels.*

This concludes the description of the algorithms for searching and updating a skip list. Observe that the word "balance" does not occur anywhere: as mentioned already, our coin takes care of this; we do not have to worry about it.

In the next sections, we will analyze the complexity of the skip list. Of course, we have to say what we mean by that. First consider the *size* of the data structure. It is clear that this size depends on the results of the coin flips that are made during the construction of the skip list. There is no *worst-case* upper bound on the size. Given any set S of real numbers, the size of a skip list for S is a *random variable*, and we are interested in the *expected value* of this variable.

The running time of the search algorithm is also a random variable. The *expected search time* is the expected value of this variable. The expectation is computed by averaging over all possible outcomes of the coin flips.

Exercise 2.5 Suppose an adversary generates search and update operations. He wants to construct a sequence of operations that takes much time to process. Assume the adversary knows the outcomes of our coin flips. Show that he can generate a sequence of update operations, followed by one search operation, such that this final operation takes linear time.

As a final remark, we have given two constructions of the sets S_i, $i \geq 1$. Both constructions define the same probability distribution on skip lists.

Therefore, if we analyze the skip list, we can use properties of both constructions. As we will see, depending on what we want to prove, the properties of one construction may be more appropriate than those of the other.

3 Why skip lists are efficient: the intuition

In this section, we give the *intuition* why skip lists are expected to be efficient. Later, we will give rigorous proofs.

First we consider the *number* of sets S_i, i.e., the value of h. According to the first construction, we get the set S_{i+1}, by taking all elements of S_i for which the coin flip produced a one. Hence, we expect that the size of S_{i+1} is about half the size of S_i. From this, we expect that the value of h is $O(\log n)$.

Let x be any element of S. How many sets S_i contain x? According to the second construction, we flip a coin until a zero comes up. We expect that this happens after a few (about two) flips. Hence, we expect that x is contained in only a small number of sets S_i. That is, any fixed element of S is expected to be stored in only a small number lists and, therefore, the size of the skip list should be $O(n)$. This also follows from the fact that $|S_{i+1}| \approx |S_i|/2$ (at least, we expect this), because this implies that the size of the skip list is proportional to $\sum_i |S_i| \approx \sum_i n/2^i = O(n)$.

Next let us consider the cost of searching for a real number x. Let C_i be the number of elements in the list L_i that are visited by the algorithm. Then the search cost is proportional to $\sum_{i=1}^{h} C_i$. Consider any fixed i. What value of C_i do we expect? Recall that y_i and y_{i+1} are the largest elements of L_i and L_{i+1} that are less than or equal to x, respectively. Let y'_{i+1} be the successor of y_{i+1} in L_{i+1}. (We assume for simplicity that y_{i+1} is not the maximal element of L_{i+1}.) Observe that $y'_{i+1} > x$. Moreover, C_i is equal to the number of elements in L_i that are to the right of y_{i+1} and to the left of y_i. (We include both y_i and y_{i+1}.) See Figure 4, where the dashed arrows form a part of the search path.

Assume that C_i is large, say 100. Our first construction algorithm implies that all coin flips for the 99 elements of L_i, from the successor of y_{i+1} up to y_i, produced a zero. Moreover, the coin flip for y_{i+1} produced a one. But this is extremely unlikely to happen: the probability is $(1/2)^{100}$. Hence, we expect C_i to be small. That is, the search cost in the list L_i should be small. Since we expect to have $O(\log n)$ such lists, the entire search algorithm should take $O(\log n)$ time. (Here, we multiply two expectations, which is in general

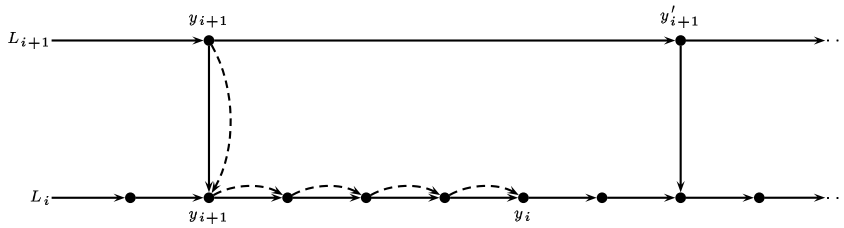

Figure 4: *The search algorithm at level i.*

---

not allowed. But remember that this section only gives intuition, not real proofs.)

The costs of the insertion and deletion algorithms are proportional to the number of lists L_i plus the cost of the search algorithm. Therefore, we also expect these costs to be $O(\log n)$.

4 Some notions from probability theory

We assume that the reader has some elementary knowledge about probability theory. We recall the basic notions.

Let U be a *sample space*. The elements of U are called *elementary events*. They can be viewed as possible outcomes of an experiment. An *event* is a subset of U. Two events A and B are called *mutually exclusive* if $A \cap B = \emptyset$.

A *probability distribution* Pr on U is a function

$$\Pr : 2^U \longrightarrow \mathbb{R}$$

that maps events to real numbers, such that

1. $\Pr(A) \geq 0$ for any event A,
2. $\Pr(U) = 1$,
3. for any finite or countably infinite sequence $A_1, A_2, \ldots$ of events that are pairwise mutually exclusive,

$$\Pr\left(\bigcup_i A_i\right) = \sum_i \Pr(A_i).$$

The real number $\Pr(A)$ is the *probability* of event A. Two events A and B are called *independent*, if $\Pr(A \cap B) = \Pr(A) \cdot \Pr(B)$.

Exercise 4.1 Prove the following statements.

1. The empty event $\emptyset$ has probability $\Pr(\emptyset) = 0$.
2. If A and B are events such that $A \subseteq B$, then $\Pr(A) \leq \Pr(B)$.
3. For any event A, $\Pr(U \setminus A) = 1 - \Pr(A)$.
4. For events A and B,

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B) \leq \Pr(A) + \Pr(B).$$

If A and B are events such that $\Pr(B) > 0$, then the *conditional probability* of event A, given that event B occurs is defined as

$$\Pr(A \mid B) := \frac{\Pr(A \cap B)}{\Pr(B)}.$$

Lemma 4.2 *Let $B_1, B_2, \ldots$ be a finite or countably infinite sequence of events that are pairwise mutually exclusive, such that $\Pr(B_i) > 0$ for all i, and $\sum_i \Pr(B_i) = 1$. Let A be any event. Then,*

$$\Pr(A) = \sum_i \Pr(A \mid B_i) \cdot \Pr(B_i).$$

Proof. Let $B := \bigcup_i B_i$ and $B^* := U \setminus B$. Since $A = (A \cap B) \cup (A \cap B^*)$, we have $\Pr(A) = \Pr(A \cap B) + \Pr(A \cap B^*)$. It follows from our assumptions that $\Pr(B) = 1$ and, hence, $\Pr(B^*) = 0$. But then, since $\Pr(A \cap B^*) \leq \Pr(B^*)$, we also have $\Pr(A \cap B^*) = 0$. Therefore,

$$\begin{aligned}
\Pr(A) &= \Pr(A \cap B) \\
&= \Pr\left(\bigcup_i (A \cap B_i)\right) \\
&= \sum_i \Pr(A \cap B_i) \\
&= \sum_i \Pr(A \mid B_i) \cdot \Pr(B_i).
\end{aligned}$$

■

Why did we define the notion of conditional probability? One reason is the following: In some applications, it is difficult to compute $\Pr(A)$ directly. In such cases, we define events B_i, $i \geq 1$, such that the conditions of Lemma 4.2 are satisfied, and such that $\Pr(A \mid B_i)$ is easy to compute. Using Lemma 4.2, this allows us to compute $\Pr(A)$.

Let U be a finite or countably infinite sample space. A *random variable* is a function $X : U \longrightarrow \mathbb{R}$. If x is a real number, then the event "$X = x$" is defined as $\{u \in U : X(u) = x\}$. This implies that

$$\Pr(X = x) = \sum_{u \in U : X(u) = x} \Pr(\{u\}).$$

Similarly, if X and Y are random variables, then for any $x, y \in \mathbb{R}$,

$$\Pr(X = x \wedge Y = y) = \sum_{u \in U : X(u) = x \wedge Y(u) = y} \Pr(\{u\}).$$

Exercise 4.3 Convince yourself that

$$\Pr(X \geq x) = \sum_{u \in U : X(u) \geq x} \Pr(\{u\}).$$

Exercise 4.4 Prove that $\Pr(X = x) = \sum_y \Pr(X = x \wedge Y = y)$. (Hint: Use Lemma 4.2. Convince yourself that the summation has a finite or countably infinite number of terms.)

The random variables X and Y are called *independent*, if for all x and y,

$$\Pr(X = x \wedge Y = y) \; = \; \Pr(X = x) \cdot \Pr(Y = y).$$

A sequence $(X_i)_{i \geq 1}$ of random variables is called *pairwise independent*, if for all i and j, $i \neq j$, and for all x and y,

$$\Pr(X_i = x \wedge X_j = y) \; = \; \Pr(X_i = x) \cdot \Pr(X_j = y).$$

The sequence is called *mutually independent*, if for all $n \geq 2$, $1 \leq i_1 < i_2 < \ldots < i_n$, and $x_1, x_2, \ldots, x_n$,

$$\Pr\left(\bigwedge_{j=1}^{n} (X_{i_j} = x_j) \right) \; = \; \prod_{j=1}^{n} \Pr(X_{i_j} = x_j).$$

Given random variables X and Y, we can define new ones, such as $X+Y$, $X \cdot Y$, and e^X, in the obvious way.

The *expected value* of a random variable X is defined as

$$E(X) := \sum_x x \cdot \Pr(X = x),$$

provided the series converges absolutely.

Exercise 4.5 Let X and Y be random variables, and let a be a real number.

1. Prove that $E(X+Y) = E(X) + E(Y)$.
2. Prove that $E(a \cdot X) = a \cdot E(X)$.
3. Assume that X and Y are independent. Prove that $E(X \cdot Y) = E(X) \cdot E(Y)$.

Exercise 4.6 Consider the following sample space:

$$U := \{(123), (132), (213), (231), (312), (321), (111), (222), (333)\}.$$

We choose a random element u from U, where each element has the same probability of being chosen. For each $i = 1, 2, 3$, let X_i be the random variable whose value is equal to the i-th number in u. For example, if $u = (312)$, then $X_3 = 2$.) Let N be the random variable, whose value is equal to that of X_2. Prove the following five claims.

1. For any i, $1 \le i \le 3$, and any r, $1 \le r \le 3$, we have $\Pr(X_i = r) = 1/3$.
2. X_1, X_2 and X_3 are pairwise independent.
3. X_1, X_2 and X_3 are *not* mutually independent.
4. $\sum_{i=1}^{E(N)} E(X_i) = 4$.
5. $E\left(\sum_{i=1}^{N} X_i\right) \neq \sum_{i=1}^{E(N)} E(X_i)$.

4.1 An example: the expected size of a random subset

Let $S = \{x_1, x_2, \ldots, x_n\}$ be a set of n elements, and let p, $0 \leq p \leq 1$, be any real number.

Consider the following experiment. For each ℓ, $1 \leq \ell \leq n$, we choose element x_ℓ with probability p. Let T be the set of all elements that are chosen, and let Y be the cardinality of T.

Observe that Y is a random variable; its value depends on the outcomes of our random choices. Our intuition says that the expected value of Y should be equal to pn. Below, we will verify this, i.e., we will prove that $E(Y) = pn$.

In a skip list, we start with a set $S_1 = S$ of size n. The second set S_2 is obtained by selecting each element of S with probability $p = 1/2$. It follows that the expected size of this second set is equal $n/2$.

4.1.1 First proof

In the first proof, we compute $E(Y)$ by "brute force", using the definition of expected value.

The random variable Y can take any value in the set $\{0, 1, 2, \ldots, n\}$. Let us fix an integer k, $0 \leq k \leq n$. What is the probability that $Y = k$? That is, what is the probability that the set T contains exactly k elements?

There are $\binom{n}{k}$ subsets of size k. Any such subset is chosen with probability $p^k(1-p)^{n-k}$. Therefore, we have

$$\Pr(Y = k) = \binom{n}{k} p^k (1-p)^{n-k}.$$

It follows that

$$\begin{aligned} E(Y) &= \sum_{k=0}^{n} k \cdot \Pr(Y = k) \\ &= \sum_{k=1}^{n} k \cdot \binom{n}{k} p^k (1-p)^{n-k} \\ &= \sum_{k=1}^{n} n \cdot \binom{n-1}{k-1} p^k (1-p)^{n-k} \\ &= pn \sum_{\ell=0}^{n-1} \binom{n-1}{\ell} p^\ell (1-p)^{n-1-\ell} \end{aligned}$$

$$
\begin{aligned}
&= pn\,(p + (1-p))^{n-1} \\
&= pn.
\end{aligned}
$$

(In the fifth line, we used Newton's binomial theorem.)

4.1.2 Second proof

We now give a simpler proof of the fact that $E(Y) = pn$. For each i, $1 \leq i \leq n$, we define a random variable Y_i, whose value is equal to

$$Y_i = \begin{cases} 1 & \text{if } x_i \in T, \\ 0 & \text{if } x_i \notin T. \end{cases}$$

Observe that the size Y of the random subset T is equal to

$$Y = \sum_{i=1}^{n} Y_i.$$

Moreover, we have

$$E(Y) = E\left(\sum_{i=1}^{n} Y_i\right) = \sum_{i=1}^{n} E(Y_i).$$

We fix a value of i, and compute the expected value of Y_i. We have

$$
\begin{aligned}
E(Y_i) &= 0 \cdot \Pr(Y_i = 0) + 1 \cdot \Pr(Y_i = 1) \\
&= \Pr(Y_i = 1) \\
&= \Pr(x_i \in T) \\
&= p.
\end{aligned}
$$

It follows that

$$E(Y) = \sum_{i=1}^{n} E(Y_i) = \sum_{i=1}^{n} p = pn.$$

Exercise 4.7 In this exercise, we introduce and analyze the *geometric distribution.* Let $0 < p < 1$ be any real number. We have a coin that comes up one with probability p, and zero with probability $1 - p$. We flip this coin independently, until a one comes up. Let X be the random variable whose value is equal to the number of times we flip the coin.

1. Prove that $\Pr(X = k) = p(1-p)^{k-1}$ for any $k \geq 1$. Any probability distribution that satisfies this equation is called a geometric distribution with parameter p.

2. Prove that $E(X) = 1/p$.

4.2 Intermezzo: a surprising application of probability theory

In this section, we show how to use probability theory in a problem that has absolutely nothing to do with probability theory.

An *embedding* of a graph G is a drawing having the following properties.

1. Vertices are drawn as points in the plane.

2. Each edge is drawn as a straight-line segment joining its two endpoints.

3. Edges do not overlap.

4. No vertex is in the interior of any edge.

5. No three or more edges have a point in common that is in the interior of each of these edges.

Two edges e and e' *cross*, if their interiors have a point in common. We call this common point a *crossing*. (See Figure 5 for an example.)

The *crossing number* $cr(G)$ of G is defined as the minimum number of crossings in any embedding of G. Hence, G is planar if and only if $cr(G) = 0$.

We consider the problem of proving a lower bound on the crossing number of graphs.

4.2.1 A simple lower bound

Let G be any graph having n vertices and m edges. Consider an embedding of G having $cr(G)$ crossings. (Hence, this embedding is the "best" one.)

We "make" G planar, by defining all crossings to be vertices. That is, let H be the graph whose vertex set is the union of the vertex set of G and the set of all crossings in the embedding. Edges of G are cut into smaller edges, which are edges in the graph H. (In Figure 6, the planar version of the graph of Figure 5 is shown.)

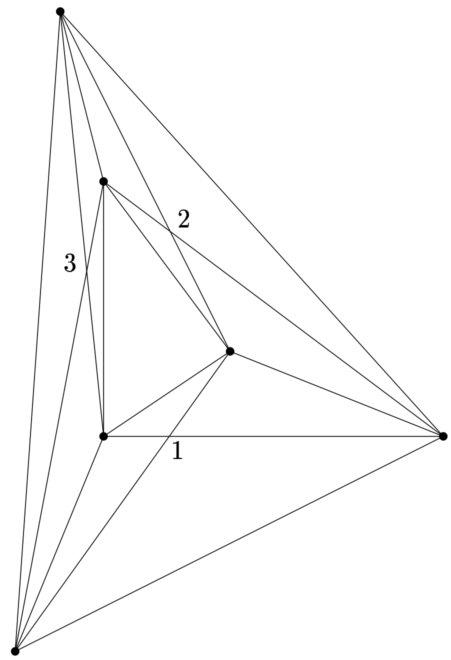

Figure 5: *An embedding of the complete graph K_6 on six vertices. In this embedding, there are three crossings.*

It is clear that the graph H is planar; we already have an embedding without any crossings. Also, this graph has $n + cr(G)$ vertices. How many edges does H have? Any crossing in G is the intersection of exactly two edges of G; these two edges contribute four smaller edges to H. Hence, for any crossing in G, the number of edges in H increases by two. It follows that H has $m + 2 \cdot cr(G)$ edges.

Since H is planar, we know that its number of edges is bounded from above by three times its number of vertices minus six, i.e.,

$$m + 2 \cdot cr(G) \leq 3(n + cr(G)) - 6,$$

which is equivalent to

$$cr(G) \geq m - 3n + 6. \tag{1}$$

Let us look at an example. For any $n \geq 1$, we denote by K_n the complete

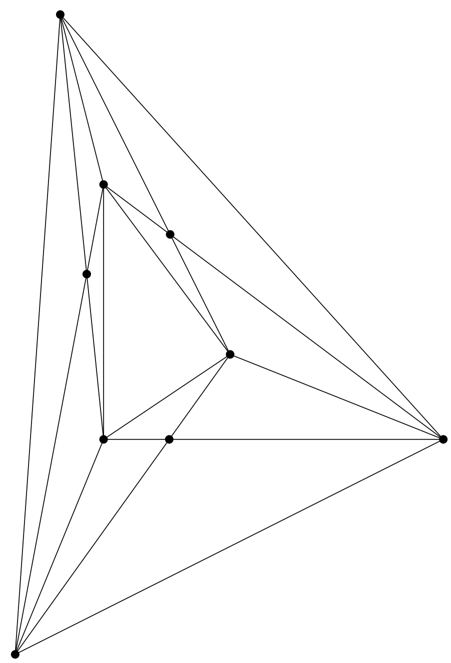

Figure 6: *The planar version of the graph of Figure 5. This planar graph has 9 vertices and 21 edges.*

---

graph on n vertices. This graph has $\binom{n}{2}$ edges. Hence,

$$cr\,(K_n) \geq \binom{n}{2} - 3n + 6 = \frac{1}{2}n^2 - \frac{7}{2}n + 6. \tag{2}$$

For $n = 6$, we get $cr\,(K_6) \geq 3$. The embedding in Figure 5 proves that $cr\,(K_6)$ is in fact equal to three.

4.2.2 A better lower bound

In this section, we will use probability theory to prove a better lower bound on the crossing number of graphs. The proof is due to Chazelle, Sharir, and Welzl. (See [1].)

As before, let G be any graph with n vertices and m edges. Again, we consider an embedding of G having $cr(G)$ crossings.

We take a real number p such that $0 < p \leq 1$. Let G_p be the random subgraph of G, that is obtained as follows. Any vertex of G appears as a vertex in G_p with probability p. Any edge of G appears as an edge in G_p if and only if both its endpoints are vertices of G_p. The embedding of G that we have fixed immediately implies an embedding of G_p. (This embedding of G_p is not necessarily the "best" one!)

We denote the number of vertices, edges, and crossings in the embedding of G_p by n_p, m_p, and x_p, respectively. Observe that these three quantities are random variables. Moreover, x_p is greater than or equal to the crossing number of the graph G_p. It follows from (1) that $cr(G_p) - m_p + 3n_p \geq 6 \geq 0$. Hence,

$$x_p - m_p + 3n_p \geq 0.$$

The left-hand side is a random variable, which is always non-negative. Therefore,

$$E\left(x_p - m_p + 3n_p\right) \geq 0.$$

Using the linearity of expectation, we get

$$E(x_p) - E(m_p) + 3 \cdot E(n_p) \geq 0. \tag{3}$$

We are now going to compute these three expected values separately.

We have seen in Section 4.1 that $E(n_p) = pn$. To compute $E(m_p)$, observe that any edge of G appears in G_p if and only if both its endpoints are vertices of G_p. It follows that any edge of G appears in G_p with probability p^2. Hence, $E(m_p) = p^2 m$. Finally, any crossing of G appears in G_p if and only if both edges that give rise to this crossing appear in G_p. This implies that any crossing of G appears in G_p with probability p^4. It follows that $E(x_p) = p^4 \cdot cr(G)$. Substituting these three expected values into (3), we get

$$p^4 \cdot cr(G) - p^2 m + 3 \cdot pn \geq 0,$$

which rewrites to

$$cr(G) \geq \frac{p^2 m - 3pn}{p^4}. \tag{4}$$

This inequality holds for *any* p, $0 < p \leq 1$.

Theorem 4.8 *Let G be any graph with n vertices and m edges. Assume that $m \geq 4n$. Then*

$$cr(G) \geq \frac{1}{64} \frac{m^3}{n^2}.$$

Proof. In (4), take $p := 4n/m$. ∎

If we apply this lower bound to the complete graph K_n, then we get $cr(K_n) = \Omega(n^4)$. This lower bound is much better than the quadratic lower bound in (2).

Remark 4.9 We now argue why the proof of Theorem 4.8 is completely shocking. Let n be a very large integer, and consider the complete graph K_n. In this case, we have $m = \binom{n}{2}$, and $p = 4n/m = 8/(n-1)$. Let us see what happens if we repeat the proof for this graph. We choose a random subgraph G_p of K_n. The expected number of vertices in G_p is equal to $pn \approx 8$, i.e., this graph is extremely small. Then we apply the *weak* lower bound (1) to this *extremely small* graph. The result is a proof that in any embedding of the *huge* graph K_n, there are $\Omega(n^4)$ crossings!

4.3 More probability theory

We will mainly use random variables that take non-negative integer values. The following lemma states how the expected value of such a random variable can be computed.

Lemma 4.10 *Let X be a random variable that takes values in $\{0, 1, 2, \ldots\}$. Then,*

$$E(X) = \sum_{\ell=1}^{\infty} \Pr(X \geq \ell).$$

Proof. Since $\Pr(X = \ell) + \Pr(X \geq \ell + 1) = \Pr(X \geq \ell)$, we get

$$\begin{aligned} E(X) &= \sum_{\ell=0}^{\infty} \ell \cdot \Pr(X = \ell) \\ &= \sum_{\ell=0}^{\infty} \ell \cdot (\Pr(X \geq \ell) - \Pr(X \geq \ell + 1)) \\ &= \sum_{\ell=1}^{\infty} \ell \cdot \Pr(X \geq \ell) - \sum_{\ell=1}^{\infty} (\ell - 1) \cdot \Pr(X \geq \ell) \\ &= \sum_{\ell=1}^{\infty} \Pr(X \geq \ell). \end{aligned}$$

■

If we want to compute the expected value of X according to the definition, then we have to know the probability that the value of X is exactly equal to k. In many applications, it is easier to compute the probability that the value of X is greater than or equal to k. In such situations, Lemma 4.10 should be applied.

Exercise 4.11 Let X and Y be random variables such that $X(u) \leq Y(u)$ for all $u \in U$.

1. Prove that $\Pr(X \geq t) \leq \Pr(Y \geq t)$ for all $t \in \mathbb{R}$.

2. Assume that X and Y take values in $\{0, 1, 2, \ldots\}$. Prove that $E(X) \leq E(Y)$.

Lemma 4.12 *Let X be a random variable and let f be any function. Then the expected value of the random variable $f(X)$ is equal to*

$$E(f(X)) = \sum_x f(x) \cdot \Pr(X = x).$$

Proof. Assume that X takes on the values $x_1, x_2, x_3, \ldots$ Let $y_1, y_2, y_3, \ldots$ be the elements of the set $\{f(x_i) : i \geq 1\}$. Hence, the y_i's are pairwise distinct. Then, by definition,

$$E(f(X)) = \sum_{i=1}^{\infty} y_i \cdot \Pr(f(X) = y_i).$$

Since

$$\Pr(f(X) = y_i) = \sum_{k : f(x_k) = y_i} \Pr(X = x_k),$$

it follows that

$$E(f(X)) = \sum_{i=1}^{\infty} \sum_{k : f(x_k) = y_i} f(x_k) \cdot \Pr(X = x_k).$$

Since $\bigcup_{i=1}^{\infty} \{k : f(x_k) = y_i\} = \{1, 2, 3, \ldots\}$ and the set on the left-hand side is a union of pairwise disjoint sets, we get

$$E(f(X)) = \sum_{j=1}^{\infty} f(x_j) \cdot \Pr(X = x_j).$$

This proves the lemma. ∎

Let X and Y be random variables. The *conditional expected value* of X, given that $Y = y$, is defined as

$$E(X \mid Y = y) := \sum_x x \cdot \Pr(X = x \mid Y = y).$$

Lemma 4.13 *We have*

$$E(X) = \sum_y E(X \mid Y = y) \cdot \Pr(Y = y).$$

Proof. We know from Lemma 4.2 that

$$\Pr(X = x) = \sum_y \Pr(X = x \mid Y = y) \cdot \Pr(Y = y).$$

Therefore,

$$\begin{aligned} E(X) &= \sum_x x \cdot \Pr(X = x) \\ &= \sum_x x \sum_y \Pr(X = x \mid Y = y) \cdot \Pr(Y = y) \\ &= \sum_y \Pr(Y = y) \sum_x x \cdot \Pr(X = x \mid Y = y) \\ &= \sum_y \Pr(Y = y) \cdot E(X \mid Y = y). \end{aligned}$$

∎

4.4 An example: the expected size of a random subset of a random subset

Let $S_1 = \{x_1, x_2, \ldots, x_n\}$ be a set of n elements, and let p, $0 \leq p \leq 1$, be any real number.

We do the following experiment. For each ℓ, $1 \leq \ell \leq n$, we choose element x_ℓ with probability p. Let S_2 be the set of all elements that are chosen, and let Y be the cardinality of S_2.

Next, we construct a random subset S_3 of S_2, by taking each element of S_2 with probability p. Let Z be the cardinality of S_3. What is the expected value of the random variable Z?

In Section 4.1, we proved that $E(Y) = pn$. Hence, *on the average*, the set S_2 contains pn elements. The set S_3 is obtained from S_2 by taking each element with probability p. Therefore, our intuition says that the expected value of Z is equal to p^2n. In this section, we will prove this formally.

How do we prove that $E(Z) = p^2n$? First note that we *cannot* apply the result of Section 4.1 to the set S_2: We proved that the expected size of a random subset of a *fixed* set of size m is equal to pm. In our case, however, the set S_2 itself is random. That is, we want to analyze the expected size of a random subset S_3 of a random subset S_2. For this, we have to be careful.

Our proof will use Lemma 4.13. According to this lemma, we have

$$E(Z) = \sum_{k=0}^{n} E(Z \mid Y = k) \cdot \Pr(Y = k).$$

We fix a value of k, and compute $E(Z \mid Y = k)$. Hence, we want to compute the expected size of the set S_3, given that the set S_2 has size exactly k. So now, the size of S_2 has been *fixed*—it is not random any more—and we can apply the result of Section 4.1. It follows that $E(Z \mid Y = k) = pk$, and we get

$$E(Z) = \sum_{k=0}^{n} pk \cdot \Pr(Y = k) \; = \; p \sum_{k=0}^{n} k \cdot \Pr(Y = k).$$

The last summation is—by definition—the expected value of the random variable Y, which we know is equal to pn. It follows that

$$E(Z) = p \cdot E(Y) = p^2 n.$$

If we apply this result to a skip list, then it follows that the expected size of the third set S_3 in this data structure is equal to $n/4$.

5 Why skip lists are efficient: the proofs

After our excursion to probability theory, we are ready to analyze skip lists rigorously. The size of a skip list and the running times of the search and update algorithms are random variables. We will prove that their expected values are bounded by $O(n)$ and $O(\log n)$, respectively.

Let us first recall the ingredients of a skip list. We have a set

$$S = \{x_1 < x_2 < x_3 < \ldots < x_n\}$$

of n elements, and construct a sequence $(S_i)_{i \geq 1}$ of random subsets, in the following way.

1. $S_1 := S$.

2. For each $i \geq 1$, S_{i+1} is obtained by choosing each element of S_i with probability $1/2$.

3. The construction stops as soon as the current set S_i is empty.

We denote the number of sets constructed by h. Hence, h is a random variable, and we have

$$\emptyset = S_h \subseteq S_{h-1} \subseteq \ldots \subseteq S_2 \subseteq S_1 = S.$$

For convenience, we define $x_0 := -\infty$.

The skip list for S is a sequence of doubly-linked lists $L_1, L_2, \ldots, L_h$, where L_i stores the elements of the set $S_i \cup \{x_0\}$, in sorted order. If $i > 1$, then each element in L_i stores a pointer to its occurrence in the list L_{i-1}.

In our proofs, we will use random variables $h(x_k)$, $1 \leq k \leq n$, where the value of $h(x_k)$ is equal to the number of lists L_i that contain x_k.

Lemma 5.1 *For any k, $1 \leq k \leq n$, and any integer $\ell \geq 1$, we have*

1. $\Pr(h(x_k) = \ell) = (1/2)^\ell$.

2. $\Pr(h(x_k) \geq \ell) = (1/2)^{\ell-1}$.

Proof. Assume that $h(x_k) = \ell$. Then element x_k occurs in the lists $L_1, L_2, \ldots, L_\ell$, and not in any of the other lists. Hence, when we determined the lists that contain x_k, our coin flips produced $\ell - 1$ ones followed by one zero. The probability that this happens is equal to $(1/2)^\ell$. This proves the first claim. The proof of the second claim is left to the reader. ∎

What is the expected number of lists that contain a fixed element x_k? The following lemma gives the answer.

Lemma 5.2 *For any k, $1 \leq k \leq n$, we have*

$$E(h(x_k)) = 2.$$

Proof. We apply Lemmas 4.10 and 5.1. Observe that we can apply Lemma 4.10, because the value of $h(x_k)$ is always a positive integer. We get

$$E(h(x_k)) = \sum_{\ell=1}^{\infty} \Pr(h(x_k) \geq \ell) = \sum_{\ell=1}^{\infty} (1/2)^{\ell-1} = 2.$$

∎

Exercise 5.3 Prove Lemma 5.2 using the definition of expected value. (This has actually already been done in Exercise 4.7.)

5.1 The expected number of lists

We prove an upper bound on the expected value of the random variable h. It is clear that

$$h = 1 + \max_{1 \leq k \leq n} h(x_k),$$

which implies that

$$E(h) = E\left(1 + \max_{1 \leq k \leq n} h(x_k)\right) = 1 + E\left(\max_{1 \leq k \leq n} h(x_k)\right).$$

It is tempting to think that $E(\max_{1 \leq k \leq n} h(x_k))$ is equal to $\max_{1 \leq k \leq n} E(h(x_k))$. This is, however, not true: By Lemma 5.2, we have

$$\max_{1 \leq k \leq n} E(h(x_k)) = \max_{1 \leq k \leq n} 2 = 2.$$

However, in Section 3, we saw that our intuition says that $E(\max_{1 \leq k \leq n} h(x_k))$ should be logarithmic in n.

Here is another example to convince you that, in general, the expected value of a maximum is not the same as the maximum of the expected values.

We have $n = 100,000$ people who play the following game. One of these people is chosen uniformly at random, i.e., each single person is chosen with probability $1/n$. The person who is chosen wins $m = 1,000$ Euro, whereas all other persons do not win anything.

For each k, $1 \leq k \leq n$, let W_k be the random variable whose value is equal to the amount of money that person k wins. Then for each k, we have

$$E(W_k) = 0 \cdot \Pr(W_k = 0) + m \cdot \Pr(W_k = m) = \frac{m}{n} = 1 \text{ Eurocent.}$$

That is, the expected amount of money that any fixed person wins is equal to one Eurocent. Hence, we have

$$\max_{1 \leq k \leq n} E(W_k) = \max_{1 \leq k \leq n} 1 \text{ Eurocent} = 1 \text{ Eurocent.}$$

Let us consider the random variable $W := \max_{1 \leq k \leq n} W_k$. Since exactly one person wins 1,000 Euro, the value of W is equal to 1,000 with probability one. Therefore,

$$E\left(\max_{1 \leq k \leq n} W_k\right) = 1,000 \text{ Euro,}$$

that is, the expected amount of money that the winner gets is equal to 1,000 Euro.

Let us go back to the problem of computing the expected value of the random variable h. For any integer $\ell \geq 0$, we have

$$h \geq \ell + 1 \text{ iff } \exists k : 1 \leq k \leq n, h(x_k) \geq \ell.$$

Using Exercise 4.1 and Lemma 5.1, we get

$$\Pr(h \geq \ell + 1) \leq \sum_{k=1}^{n} \Pr(h(x_k) \geq \ell) = \sum_{k=1}^{n} (1/2)^{\ell - 1} = n\,(1/2)^{\ell - 1}. \qquad (5)$$

Observe that this estimate does not make sense if $\ell \leq \lceil \log n \rceil$. For such values of ℓ, we can use the trivial upper bound $\Pr(h \geq \ell + 1) \leq 1$. Applying Lemma 4.10, we get

$$E(h) = \sum_{\ell=0}^{\infty} \Pr(h \geq \ell + 1) = \sum_{\ell=0}^{\lceil \log n \rceil} \Pr(h \geq \ell + 1) + \sum_{\ell=1+\lceil \log n \rceil}^{\infty} \Pr(h \geq \ell + 1).$$

The first summation on the right-hand side is less than or equal to $1 + \lceil \log n \rceil$. The second summation is bounded from above by

$$\sum_{\ell=1+\lceil \log n \rceil}^{\infty} n\,(1/2)^{\ell - 1} = n\,(1/2)^{\lceil \log n \rceil - 1} \leq n\,(1/2)^{\log n - 1} = 2.$$

Hence, we have proved the following result.

Lemma 5.4 *The expected value of the number h of lists in a skip list for n elements satisfies*

$$E(h) \leq 3 + \lceil \log n \rceil.$$

Exercise 5.5 We are given n buckets and n balls. We distribute these balls over the buckets according to a uniform distribution. For any i, $1 \leq i \leq n$, let X_i be the random variable whose value is equal to the number of balls in the i-th bucket.

1. Let i be a fixed integer such that $1 \leq i \leq n$. Prove that $E(X_i) = 1$. Hence, any fixed bucket contains—expected—exactly one ball.

2. Let $X := \max_{1 \leq i \leq n} X_i$. Hence, X is the number of balls in the "largest" bucket. Prove the following claims.

   (a) $\Pr(X \geq k) \leq n \cdot \binom{n}{k} \cdot (1/n)^k$.

   (b) $E(X) \leq \sum_{k=1}^{\infty} \min(1, n/k!)$.

   (c) $E(X) = O(\log n / \log\log n)$.

   (Gonnet [4] has shown that $E(X) = \Theta(\log n / \log\log n)$. Hence, the largest bucket contains—expected—about $\log n / \log\log n$ balls.)

5.2 The expected size of a skip list

Let M be the random variable whose value is equal to the total size of the sets $S_1, S_2, \ldots, S_h$. Hence, since $S_h = \emptyset$, we have $M = \sum_{i=1}^{h-1} |S_i|$. We want to compute the expected value of M, i.e.,

$$E(M) = E\left(\sum_{i=1}^{h-1} |S_i|\right).$$

Is this summation equal to $\sum_{i=1}^{h-1} E(|S_i|)$? Equivalently, can we apply the linearity of expectation? The answer is "no": The expected value of the summation is a fixed real number. On the other hand, the summation of the expected values is a random variable; it depends on h, whose value can be *any* positive integer.

We can apply the linearity of expectation, if the number of terms in the summation is *fixed*, i.e., does not depend on any random experiment. (See also Exercise 4.6.)

We can compute the expected value of M, by observing that $M = \sum_{k=1}^{n} h(x_k)$. This is again a sum of random variables, but now the number of terms is fixed. Therefore, we can apply the linearity of expectation to this sum, and get

$$E(M) = E\left(\sum_{k=1}^{n} h(x_k)\right) = \sum_{k=1}^{n} E(h(x_k)) = \sum_{k=1}^{n} 2 = 2n.$$

We denote the total number of nodes in the skip list by M'. Hence, M' is equal to M (= the number of nodes in the lists $L_i \setminus \{-\infty\}$, $1 \leq i \leq h$) plus h (= the number of occurrences of $-\infty$). It follows that

$$E(M') = E(M + h) = E(M) + E(h) \leq 2n + 3 + \lceil \log n \rceil.$$

Since each node of the skip list contains a constant amount of information (one element of $S \cup \{-\infty\}$, and at most three pointers), this proves that its expected size is bounded by $O(n)$. We summarize this result.

Lemma 5.6 *The expected number of nodes in a skip list for n elements is less than or equal to $2n + 3 + \lceil \log n \rceil$.*

Exercise 5.7 Let $(X_i)_{i \geq 1}$ be a sequence of random variables that are distributed according to the same probability distribution. Assume that each variable X_i only takes non-negative integers as value. Let N be any random variable that also takes non-negative integers as value.

Assume that the sequence $N, X_1, X_2, \ldots$ is mutually independent. Prove that

$$E\left(\sum_{i=1}^{N} X_i\right) = E(N) \cdot E(X_1).$$

5.3 The expected search time

In this section, we prove that expected time to search in a skip list is bounded by $O(\log n)$. Recall that we numbered the elements of the set S as

$$S = \{x_1 < x_2 < x_3 < \ldots < x_n\}.$$

We define $x_0 := -\infty$ and $x_{n+1} := \infty$.

For any real number x, let $path(x)$ denote the path in the skip list that the algorithm follows when searching for x. We denote the number of nodes on this path by $P(x)$. Hence, we want to estimate the expected value of the random variable $P(x)$.

Consider any real number x. Let m be the index, such that $x_m \leq x < x_{m+1}$. Observe that $0 \leq m \leq n$. When we search for x, we find the largest element of S that is less than or equal to x. It follows that $path(x) = path(x_m)$.

Hence, in the following, we fix an integer m, such that $0 \leq m \leq n$, and analyze the path $path(x_m)$ and the number $P(x_m)$ of nodes it contains.

For each $i \geq 1$, let C_i denote the number of nodes of the list L_i that are on $path(x_m)$. Then

$$P(x_m) = \sum_{i=1}^{h} C_i.$$

Since the number of terms in this summation is a random variable, the value of $E(P(x_m))$ is *not* equal to $\sum_{i=1}^{h} E(C_i)$.

On the other hand, each of the variables C_i, $i \geq 1$, is "more or less" distributed according to a geometric distribution wit parameter $p = 1/2$. Hence, we may hope that Exercises 5.7 and 4.7 can be used to compute $E(P(x_m))$. Unfortunately, this is not the case, because the sequence $h, C_1, C_2, \ldots$, is *not* mutually independent. In fact, this sequence of random variables isn't even pairwise independent: It is easy to see that

$$\Pr(C_{i-1} = 3n/4 \wedge C_i = 3n/4) = 0,$$

whereas

$$\Pr(C_{i-1} = 3n/4) \cdot \Pr(C_i = 3n/4) \neq 0.$$

We proceed as follows. We write

$$P(x_m) = \sum_{i=1}^{\lceil \log n \rceil} C_i + \sum_{i=1+\lceil \log n \rceil}^{h} C_i,$$

i.e., we write $P(x_m)$ as a sum of two summations. The first summation has $\lceil \log n \rceil$ (i.e., a fixed number of) terms. Therefore, we have

$$E(P(x_m)) = \sum_{i=1}^{\lceil \log n \rceil} E(C_i) + E\left(\sum_{i=1+\lceil \log n \rceil}^{h} C_i \right). \tag{6}$$

We will analyze each of the expected values on the right-hand side separately.

5.3.1 The expected search time above level $\lceil \log n \rceil$

In this section, we prove an upper bound on $E\left(\sum_{i=1+\lceil \log n \rceil}^{h} C_i\right)$. We first observe that the summation is non-zero if and only if $h \geq 1 + \lceil \log n \rceil$. Since h is random, we apply Lemma 4.13: We define a random variable H whose value is equal to

$$H = \begin{cases} 1 & \text{if } h \geq 1 + \lceil \log n \rceil, \\ 0 & \text{if } h \leq \lceil \log n \rceil. \end{cases}$$

Then, by Lemma 4.13, we get

$$\begin{aligned} E\left(\sum_{i=1+\lceil \log n \rceil}^{h} C_i\right) &= E\left(\sum_{i=1+\lceil \log n \rceil}^{h} C_i \mid H = 1\right) \cdot \Pr(H = 1) \\ &\quad + E\left(\sum_{i=1+\lceil \log n \rceil}^{h} C_i \mid H = 0\right) \cdot \Pr(H = 0). \end{aligned}$$

The second summation on the right-hand side is zero: if $H = 0$, then $h \leq \lceil \log n \rceil$ and, therefore, $\sum_{i=1+\lceil \log n \rceil}^{h} C_i = 0$. This, together with the fact that $\Pr(H = 1) \leq 1$, implies that

$$E\left(\sum_{i=1+\lceil \log n \rceil}^{h} C_i\right) \leq E\left(\sum_{i=1+\lceil \log n \rceil}^{h} C_i \mid H = 1\right).$$

So assume that $H = 1$, i.e., $h \geq 1 + \lceil \log n \rceil$. The summation on the right-hand side counts the total number of nodes in the lists L_i, $1 + \lceil \log n \rceil \leq i \leq h$ that are visited when searching for element x_m. Consider the part of $path(x_m)$ that is contained in these lists. In each node on this path, we either make one steps downwards, or we make one step to the right.

The number of nodes in which we make one step downwards is equal to $h - \lceil \log n \rceil$. The elements of $S \cup \{-\infty\}$ that are stored in the nodes in which we make one step to the right are pairwise distinct. Hence, the number of nodes in which we make one step to the right is bounded from above by one[1] plus the number of elements of S that are stored in the lists L_i, $1 + \lceil \log n \rceil \leq i \leq h$, and that are less than or equal to x_m. Let Z be the number of these elements. That is, Z is the random variable whose value is equal to the number of indices k, $1 \leq k \leq m$, for which $h(x_k) \geq 1 + \lceil \log n \rceil$. Then

$$\begin{aligned} E\left(\sum_{i=1+\lceil \log n \rceil}^{h} C_i \mid H = 1 \right) &\leq E(h - \lceil \log n \rceil + 1 + Z) \\ &= E(h) - \lceil \log n \rceil + 1 + E(Z) \\ &\leq 4 + E(Z), \end{aligned}$$

where the last inequality follows from Lemma 5.4.

So it remains to estimate $E(Z)$. For each k, $1 \leq k \leq m$, we define a random variable Z_k whose value is equal to

$$Z_k = \begin{cases} 1 & \text{if } h(x_k) \geq 1 + \lceil \log n \rceil, \\ 0 & \text{if } h(x_k) \leq \lceil \log n \rceil. \end{cases}$$

Then $Z = \sum_{k=1}^{m} Z_k$, which is a sum of a fixed number of terms. Hence,

$$\begin{aligned} E(Z) &= \sum_{k=1}^{m} E(Z_k) \\ &= \sum_{k=1}^{m} \Pr(Z_k = 1) \\ &= \sum_{k=1}^{m} \Pr(h(x_k) \geq 1 + \lceil \log n \rceil) \\ &= \sum_{k=1}^{m} (1/2)^{\lceil \log n \rceil} \\ &\leq \sum_{k=1}^{m} 1/n \\ &= m/n \\ &\leq 1. \end{aligned}$$

[1]this is the last occurrence of $-\infty$ that we visit or, equivalently, the occurrence of $-\infty$ in which we make one step to the right

Overall, we have proved the following upper bound on the expected search time in the lists L_i, $1 + \lceil \log n \rceil \leq i \leq h$:

$$E\left(\sum_{i=1+\lceil \log n \rceil}^{h} C_i\right) \leq 5.$$

5.3.2 The expected search time up to level $\lceil \log n \rceil$

We now analyze the summation $\sum_{i=1}^{\lceil \log n \rceil} E(C_i)$.

Consider any integer i, such that $1 \leq i \leq \lceil \log n \rceil$. Recall that C_i is the number of nodes in list L_i that are on the search path to element x_m. In Section 3, we argued that our intuition tells us that C_i is distributed according to a geometric distribution with parameter $p = 1/2$. (See Exercise 4.7.) If this were true, then it would be easy to compute our summation.

Unfortunately, the random variable C_i is *not* distributed according to a geometric distribution with parameter $p = 1/2$: For any $k \geq m + 2$, we have $\Pr(C_i = k) = 0$, because we make at most $m + 1$ steps in list L_i. In a geometric distribution with parameter $p = 1/2$, this probability must be equal to $(1/2)^k$.

So, how do we proceed? Well, we will prove that each random variable C_i is bounded from above by a random variable Y_i which is distributed according to a geometric distribution with parameter $p = 1/2$. This will enable us to estimate $\sum_{i=1}^{\lceil \log n \rceil} E(C_i)$.

Consider the following alternative construction of our skip list. We rename the elements of S as

$$S = \{s_{11} < s_{12} < \ldots < s_{1n}\}.$$

Let $S_1 := S$, and $n_1 := n$. Moreover, let $\ell_1(x_m)$ be the number of elements of S_1 that are less than or equal to x_m.

Let $i \geq 1$, and assume that we have constructed the set

$$S_i = \{s_{i1} < s_{i2} < \ldots < s_{in_i}\},$$

where $n_i = |S_i|$. Let

$$\ell_i(x_m) := |\{y \in S_i : y \leq x_m\}|.$$

We do the following.

Step 1: We flip our coin $n_i - \ell_i(x_m)$ times.

Step 2: We flip our coin $\ell_i(x_m)$ times.

Step 3: If we only got zeroes in Step 2, then we keep on flipping our coin until we get a one.

Let f_i be the total number of flips made during these three steps, and denote the outcomes by

$$F_{i1}, F_{i2}, \ldots, F_{if_i}.$$

Observe that $f_i \geq n_i$. We proceed as follows:

1. We define
$$S_{i+1} := \{s_{ij} : 1 \leq j \leq n_i \text{ and } F_{ij} = 1\},$$
$n_{i+1} := |S_{i+1}|$, and $\ell_{i+1}(x_m) := |\{y \in S_{i+1} : y \leq x_m\}|$.

2. We define a random variable Y_i whose value is equal to the total number of times we flipped our coin in Steps 2 and 3, until we got a one for the first time.

3. We define a random variable C'_i whose value is equal to
$$C'_i = \min(1 + \ell_i(x_m), Y_i).$$

The construction stops as soon as the current set S_i is empty. We denote the number of sets by h. Hence, we get a sequence

$$\emptyset = S_h \subseteq S_{h-1} \subseteq S_{h-2} \subseteq \ldots \subseteq S_2 \subseteq S_1 = S$$

of random subsets of S, and two sequences $C'_1, C'_2, \ldots, C'_h$, and $Y_1, Y_2, \ldots, Y_h$ of random variables.

The skip list for S is now obtained by storing each set $S_i \cup \{-\infty\}$, $1 \leq i \leq h$, in a list L_i, as before.

This new construction defines a probability distribution on skip lists, which is the same as that of our previous two constructions. In fact, the new construction is the same as our first one, except that Step 3 and the random variables Y_i and C'_i, $i \geq 1$, have been added.

Consider any integer $i \geq 1$. Then C'_i is equal to the number of nodes in list L_i that we visit when searching for element x_m. This random variable C'_i

is distributed according to the same probability distribution as our random variable C_i. Therefore, we have

$$\sum_{i=1}^{\lceil \log n \rceil} E(C_i) = \sum_{i=1}^{\lceil \log n \rceil} E(C_i').$$

Next, we observe that for any $i \geq 1$,

1. $C_i' \leq Y_i$, and
2. Y_i is distributed according to a geometric distribution with parameter $p = 1/2$.

Hence, by Exercise 4.7, we have $E(Y_i) = 2$ for any $i \geq 1$. Also, by Exercise 4.11, we have $E(C_i') \leq E(Y_i) = 2$. It follows that

$$\sum_{i=1}^{\lceil \log n \rceil} E(C_i) \leq \sum_{i=1}^{\lceil \log n \rceil} 2 = 2\lceil \log n \rceil.$$

5.3.3 The overall expected search time

If we combine the results of Sections 5.3.1 and 5.3.2 with Equation (6), then we get the following upper bound on the expected number $E(P(x_m))$ of nodes on the search path to element x_m:

$$E(P(x_m)) \leq 2\lceil \log n \rceil + 5.$$

Hence, we have proved that the expected time to search for any fixed element x_m is bounded by $O(\log n)$. We have seen already in the beginning of Section 5.3 that this implies that the expected time to search for any fixed real number x is also bounded $O(\log n)$.

Let us consider the expected time to insert an element x into the skip list. Assume for simplicity that x is not in the current set S. Hence x is a new element. When we insert x, we first search for the largest element that is smaller than x. We have seen that this takes $O(\log n)$ expected time. Then we flip our coin until we get a zero. The number of flips determines the number of lists L_i in which x gets inserted. Since any fixed element occurs—expected—in two lists, this final step of the insertion algorithm takes $O(1)$ expected time. Hence, by using linearity of expectation, we have shown that the expected time to insert an element into a skip list is bounded by $O(\log n) + O(1) = O(\log n)$.

It is clear that the time to delete any fixed element from a skip list is proportional to the time to search for this element. Therefore, the expected deletion time is also bounded by $O(\log n)$.

Here is an important remark. An insertion (resp. deletion) of an element x results in a skip list that comes from the *same* probability distribution as a skip list for $S \cup \{x\}$ (resp. $S \setminus \{x\}$) that is built by any one of our three constructions. Hence, after a sequence of updates has been performed, the data structure still "behaves" as if it were just built "from scratch". We analyzed skip lists under the assumption that they were built by one of our three constructions. As a result, the expected space and time bounds we derived also hold for skip lists that have been changed by a sequence of updates.

We can now state the results of this section.

Theorem 5.8 *Let S be a set of n real numbers and let SL be a skip list for S.*

1. *The expected number of lists in SL is less than or equal to* $3 + \lceil \log n \rceil$.
2. *The expected size of SL is bounded by* $O(n)$.
3. *For any fixed $x \in \mathbb{R}$, it takes $O(\log n)$ expected time to search in SL for the largest element of S that is less than or equal to x.*
4. *We can insert any fixed element into SL in $O(\log n)$ expected time.*
5. *We can delete any fixed element from SL in $O(\log n)$ expected time.*

6 Tail estimates: Chernoff bounds

In the previous section, we proved bounds on the expected size, search time and update time of a skip list. In this section, we consider so-called *tail estimates*. That is, we estimate the probability that, e.g., the *actual* search time deviates significantly from its expected value.

We saw that the expected search time is bounded by $O(\log n)$. Assume for simplicity that the constant in this bound is equal to one. Then we want to estimate the probability that the actual search time is greater than or equal to $t \cdot \log n$ for some large value of t. This probability can be bounded by *Markov's inequality*:

Lemma 6.1 *Let X be any random variable that takes non-negative values, and let μ be the expected value of X. Then for any $t > 0$, we have*

$$\Pr(X \geq t\mu) \leq 1/t.$$

Proof. Let $s := t\mu$. Then $s \geq 0$, and

$$\begin{aligned} \mu &= E(X) \\ &= \sum_x x \cdot \Pr(X = x) \\ &\geq \sum_{x \geq s} x \cdot \Pr(X = x) \\ &\geq \sum_{x \geq s} s \cdot \Pr(X = x) \\ &= s \cdot \Pr(X \geq s). \end{aligned}$$

∎

Let us apply this inequality to the random variable $P(x_m)$, which counts the number of nodes in the skip list that are visited when searching for element x_m. So, $\mu = E(P(x_m))$. We have seen that

$$\mu \leq 2\lceil \log n \rceil + 5.$$

Let $t > 0$. Then, if $P(x_m) \geq t(2\lceil \log n \rceil + 5)$, then $P(x_m) \geq t\mu$. It follows that

$$\Pr(P(x_m) \geq t(2\lceil \log n \rceil + 5)) \leq \Pr(P(x_m) \geq t\mu) \leq 1/t.$$

This inequality is not very impressive: It shows that the probability that the actual search time is at least 100 times its expected value is less than or equal to 1/100. If this upper bound were tight, then among 100 searches we would expect that one takes 100 times as long as an average search.

In this section, we prove so-called *Chernoff bounds*, which will be used in Section 7 to give much better tail estimates. By using more properties of the random variables that determine the search time, we will prove that the probability that it exceeds $t \cdot \log n$ is less than or equal to $n^{-9t/50}$, for $t \geq 5$.

Hence, in a skip list for 1000 elements, the probability that the search time is more than 100 times its expected value is less than or equal to 10^{-54}. In practice, this means that this event will *never* occur.

Markov's inequality holds for any non-negative random variable. The Chernoff technique applies to random variables X that can be written as a sum $\sum_{i=1}^{n} X_i$ of mutually independent random variables X_i. In such cases, much better bounds can be obtained for $\Pr(X \geq t\mu)$.

So let $X_1, X_2, \ldots, X_n$ be a sequence of n mutually independent random variables, and let $X = \sum_{i=1}^{n} X_i$. For any real number λ, the random variables $e^{\lambda X_1}, e^{\lambda X_2}, \ldots, e^{\lambda X_n}$ are also mutually independent. Therefore,

$$E\left(e^{\lambda X}\right) = E\left(e^{\lambda(X_1+\ldots+X_n)}\right) = E\left(\prod_{i=1}^{n} e^{\lambda X_i}\right) = \prod_{i=1}^{n} E\left(e^{\lambda X_i}\right).$$

Now let $s > 0$ and $\lambda > 0$. Since $X \geq s$ if and only if $e^{\lambda X} \geq e^{\lambda s}$, we have

$$\Pr(X \geq s) = \Pr\left(e^{\lambda X} \geq e^{\lambda s}\right).$$

By applying Markov's inequality to the non-negative random variable $e^{\lambda X}$, we get

$$\Pr(X \geq s) = \Pr\left(e^{\lambda X} \geq e^{\lambda s}\right) \leq e^{-\lambda s} \cdot E\left(e^{\lambda X}\right).$$

This implies that

$$\Pr(X \geq s) \leq e^{-\lambda s} \prod_{i=1}^{n} E\left(e^{\lambda X_i}\right) \text{ for } s > 0 \text{ and } \lambda > 0. \tag{7}$$

This is the basic inequality that we will use. It tells us that to estimate $\Pr(X \geq s)$, we need bounds on $E(e^{\lambda X_i})$. Of course, these bounds depend on the probability distribution of X_i.

We illustrate the technique for the *geometric distribution.* (See Exercise 4.7.) We are given a coin that comes up with zero or one, each with probability 1/2. We flip this coin independently until a one comes up. Let T be the number of flips. Then T is distributed according to a geometric distribution with parameter $p = 1/2$, i.e., $\Pr(T = k) = (1/2)^k$ for any $k \geq 1$, and $E(T) = 2$.

Now assume that we flip the coin until we have obtained exactly n ones. We denote the total number of flips by T_n. (Hence, $T = T_1$.) This random variable T_n is distributed according to a *negative binomial distribution.* Our goal is to estimate $\Pr(T_n \geq s)$.

To apply the Chernoff technique, we have to express T_n as a sum of mutually independent random variables. For each i, $1 \le i \le n$, let X_i denote the number of flips between the $(i-1)$-st one and the i-th one. (In this count, we exclude the flip that gives the $(i-1)$-st one, but we include the flip that gives the i-th one.) Then, $T_n = \sum_{i=1}^{n} X_i$, each X_i is distributed according to a geometric distribution with parameter $p = 1/2$, and the variables $X_1, X_2, \ldots, X_n$ are mutually independent.

The expected value of T_n follows from the linearity of expectation:

$$E(T_n) = \sum_{i=1}^{n} E(X_i) = \sum_{i=1}^{n} 2 = 2n.$$

Hence Markov's inequality gives

$$\Pr(T_n \ge (2+t)n) \le 2/(2+t). \tag{8}$$

As we will see, the Chernoff technique gives a much better upper bound.

Let λ be such that $0 < \lambda < \ln 2$. Then, applying Lemma 4.12 with $f(x) = e^{\lambda x}$, we get for any i, $1 \le i \le n$,

$$E\left(e^{\lambda X_i}\right) = \sum_{k=1}^{\infty} e^{\lambda k} \cdot \Pr(X_i = k) = \sum_{k=1}^{\infty} (e^{\lambda}/2)^k = \frac{e^{\lambda}}{2 - e^{\lambda}}.$$

Now we apply the basic inequality (7), with $s = (2+t)n$, where $t > 0$. We get

$$\Pr(T_n \ge (2+t)n) \le e^{-\lambda(2+t)n} \left(\frac{e^{\lambda}}{2-e^{\lambda}}\right)^n = \left(\frac{e^{-\lambda(1+t)}}{2-e^{\lambda}}\right)^n.$$

This inequality holds for any λ, $0 < \lambda < \ln 2$. Hence, we now choose λ such that the term on the right-hand side is minimum. A straightforward calculation shows that this term is minimum for $\lambda = \ln(1 + \frac{t}{2+t})$. Hence,

$$\Pr(T_n \ge (2+t)n) \le \left(1 + \frac{t}{2}\right)^n \left(1 - \frac{t}{2+2t}\right)^{(1+t)n}.$$

Since $1 - x \le e^{-x}$ for all x, we have

$$\left(1 - \frac{t}{2+2t}\right)^{1+t} \le \left(e^{-t/(2+2t)}\right)^{1+t} = e^{-t/2}.$$

Moreover, $1 + t/2 \leq e^{t/4}$ for $t \geq 3$. This proves that for $t \geq 3$, we have

$$\Pr(T_n \geq (2+t)n) \leq e^{tn/4} \cdot e^{-tn/2} = e^{-tn/4}.$$

Compare this with the bound (8) obtained using Markov's inequality! Hence, we have proved the following theorem.

Theorem 6.2 *Let $X_1, X_2, \ldots, X_n$ be mutually independent random variables, each one of which is distributed according to a geometric distribution with parameter $p = 1/2$. Let $T_n = X_1 + X_2 + \ldots + X_n$. Then $E(T_n) = 2n$, and for any $t \geq 3$,*

$$\Pr(T_n \geq (2+t)n) \leq e^{-tn/4}.$$

Corollary 6.3 *Let $c \geq 1$ be any constant, and let n be any positive integer. Then for any $s \geq 5$,*

$$\Pr\left(T_{\lceil c \cdot \ln n \rceil} \geq s \cdot c \cdot \ln n\right) \leq n^{-(s-2)c/4}.$$

7 Tail estimates for skip lists

We use the results of the previous section to prove tail estimates for the size, search time and update time of a skip list.

Consider a skip list for a set S of n elements. Let h denote the number of lists L_i, let M denote the total size of the sets $S_1, S_2, \ldots, S_h$, and let M' denote the total number of nodes of the skip list. Hence, $M' = M + h$. We have seen that the expected number of nodes of the skip list is equal to $E(M') = E(M) + E(h) \leq 2n + 3 + \lceil \log n \rceil$. We are interested in the probability that M' is greater than or equal to $(2+t)n$, for a large value of t.

Clearly, if $M' \geq (2+t)n$, then $h \geq tn/2$ or $M \geq (2 + t/2)n$. As a result,

$$\Pr(M' \geq (2+t)n) \leq \Pr(h \geq tn/2) + \Pr(M \geq (2+t/2)n).$$

We know from (5), that $\Pr(h \geq \ell + 1) \leq n\,(1/2)^{\ell-1}$ for any $\ell \geq 0$. Hence, for $t \geq 1$ and n sufficiently large, we have

$$\Pr(h \geq tn/2) \leq n \left(\frac{1}{2}\right)^{tn/2-2} = e^{\ln n + 2\ln 2 - (tn/2)\ln 2} \leq e^{-tn/8}.$$

It remains to bound $\Pr(M \geq (2+t/2)n)$. For any element $x \in S$, let $h(x)$ be the number of sets S_i, $1 \leq i \leq h$, that contain x. These random variables $h(x)$, $x \in S$, are mutually independent, and each one of them is distributed according to a geometric distribution with $p = 1/2$. Since $M = \sum_{x \in S} h(x)$, Theorem 6.2 implies that for $t \geq 6$,

$$\Pr(M \geq (2 + t/2)n) \leq e^{-tn/8}.$$

This proves that for $t \geq 6$,

$$\Pr(M' \geq (2 + t)n) \leq 2 \cdot e^{-tn/8},$$

i.e., it is extremely unlikely that the size of a skip list deviates significantly from its expected value.

Let us now consider the search time. Let $x \in \mathbb{R}$, and let $c \geq 1$ be a constant. We want to estimate the probability that the algorithm that searches for x visits more than $c \cdot \ln n$ nodes. We analyze this probability by considering the nodes up to level $\lceil c \cdot \ln n \rceil$ and above level $\lceil c \cdot \ln n \rceil$ separately.

Let T_a denote the total number of nodes in the lists L_i, $1 + \lceil c \cdot \ln n \rceil \leq i \leq h$, that are visited when searching for x.

Exercise 7.1 Prove that the expected value of the random variable T_a is bounded by

$$E(T_a) = O(n^{2-2c\ln 2}). \qquad (9)$$

(*Hint:* The proof is similar to the one in Section 5.3.1.)

If we apply Markov's inequality to (9), then we get

$$\Pr(T_a \geq 1) \leq E(T_a) = O(n^{1-2c\ln 2}).$$

Let T_b denote the total number of nodes in the lists L_i, $1 \leq i \leq \lceil c \cdot \ln n \rceil$, that are visited when searching for x. In order to apply Corollary 6.3, we have to express T_b as a sum of mutually independent random variables, each one distributed according to a geometric distribution with parameter $1/2$.

For each i, $1 \leq i \leq h$, let C_i be the number of nodes in list L_i that are visited when searching for element x. For convenience, we define $C_i = 0$ for all $i > h$. We have

$$T_b = \sum_{i=1}^{\lceil c \cdot \ln n \rceil} C_i.$$

We have seen already that none of the random variables C_i is distributed according to a geometric distribution with parameter 1/2. Also, the sequence $C_1, C_2, \ldots, C_{\lceil c \cdot \ln n \rceil}$ is not mutually independent.

Consider the random variables Y_i, $1 \leq i \leq h$, that we used in Section 5.3.2. These variables are distributed according to a geometric distribution with parameter 1/2. For each i, $h < i \leq \lceil c \cdot \ln n \rceil$, we define Y_i to be a random variable that is also distributed according to this distribution. Then, we have $C_i \leq Y_i$ for all i, $1 \leq i \leq \lceil c \cdot \ln n \rceil$. Moreover, the sequence Y_i, $1 \leq i \leq \lceil c \cdot \ln n \rceil$, is mutually independent. Since

$$T_b = \sum_{i=1}^{\lceil c \cdot \ln n \rceil} C_i \leq \sum_{i=1}^{\lceil c \cdot \ln n \rceil} Y_i,$$

it follows that

$$\Pr(T_b \geq s \cdot c \cdot \ln n) \leq \Pr\left(\sum_{i=1}^{\lceil c \cdot \ln n \rceil} Y_i \geq s \cdot c \cdot \ln n \right).$$

Then Corollary 6.3 immediately implies that

$$\Pr(T_b \geq s \cdot c \cdot \ln n) \leq n^{-(s-2)c/4} \leq n^{-sc/8},$$

for $c \geq 1$ and $s \geq 5$.

Now we can complete the tail estimate for the search time. Let $P(x)$ denote the total number of nodes in the skip list that are visited when searching for x. Then $P(x) = T_a + T_b$. Moreover, if $P(x) \geq 1 + 5c \cdot \ln n$, then $T_a \geq 1$ or $T_b \geq 5c \cdot \ln n$. Hence,

$$\Pr(P(x) \geq 1 + 5c \cdot \ln n) \leq \Pr(T_a \geq 1) + \Pr(T_b \geq 5c \cdot \ln n).$$

Our results for the two probabilities on the right-hand side imply that for $c \geq 1$,

$$\Pr(P(x) \geq 1 + 5c \cdot \ln n) = O\left(n^{1-2c \cdot \ln 2} + n^{-5c/8}\right).$$

Taking $c = t/(5 \ln 2)$, where $t \geq 5 \ln 2 \approx 3.47$, we get

$$\begin{aligned} \Pr(P(x) \geq 1 + t \log n) &= O\left(n^{1-2t/5} + n^{-t/(8 \ln 2)}\right) \\ &= O\left(n^{1-2t/5} + n^{-9t/50}\right). \end{aligned}$$

This completes the analysis of the search time. Since the update time is proportional to the search time, similar bounds can be proved for it. We summarize our result.

Theorem 7.2 *Let S be a set of n real numbers, and let SL be a skip list for S.*

1. *For any $t \geq 6$, the probability that SL has more than $(2+t)n$ nodes is less than or equal to $2 \cdot e^{-tn/8}$.*
2. *For any $x \in \mathbb{R}$ and $t \geq 5 \ln 2$, the probability that a search in SL for x visits more than $1 + t \log n$ nodes is bounded by*
$$O\left(n^{1-2t/5} + n^{-9t/50}\right).$$
3. *There is a constant c such that for any sufficiently large t, the probability that an inserton or deletion in SL visits more than $t \log n$ nodes is bounded by*
$$O\left(n^{-ct}\right).$$

Remark 7.3 Consider again the random variables C_i and Y_i. We have seen that the total number $P(x)$ of nodes visited when searching for x is equal to

$$P(x) = \sum_{i=1}^{h} C_i \leq \sum_{i=1}^{h} Y_i.$$

Therefore, the expected search time $E(P(x))$ is bounded from above by $E(\sum_{i=1}^{h} Y_i)$. (See Exercise 4.11.) The sequence $h, Y_1, Y_2, \ldots$ of random variables is mutually independent. (Convince yourself that this is true.) Therefore, by Exercise 5.7,

$$E(P(x)) \leq E\left(\sum_{i=1}^{h} Y_i\right) = E(h) \cdot E(Y_1) \leq (3 + \lceil \log n \rceil) \cdot 2 = O(\log n).$$

This gives an alternative proof of the fact that the expected search time is logarithmic in n.

Exercise 7.4 Let S be a set of n real numbers, and let SL be a skip list for S. For any $x \in \mathbb{R}$, let $T(x)$ be the time to search in SL for element x. We have seen that the expected value of $T(x)$ is bounded by $O(\log n)$. Define a random variable T whose value is equal to $T = \max\{T(x) : x \in \mathbb{R}\}$. Prove that the expected value of T is also bounded by $O(\log n)$.

Exercise 7.5 Consider a skip list for a set S of n elements. For any x and y in S, $x < y$, let $R(x,y)$ be the number of elements $z \in S$ such that $x \leq z \leq y$. Assume that we know the position of x in the list L_1. Prove that the position of y in L_1 can be found in $O(\log R(x,y))$ expected time.

8 An alternative analysis of the expected search time

In this final section, we give another proof of the fact that the expected time to search in a skip list is logarithmic. This proof uses less probability theory, but more "brute-force" analysis.

Consider a skip list for a set S of n real numbers, and let $x \in \mathbb{R}$. We analyze the expected time for searching the largest element in the set $S \cup \{-\infty\}$ that is less than or equal to x. We recall the following facts.

Lemma 8.1 *For any $y \in S$, let $h(y)$ be the number of sets S_i that contain y. Then for any integer $k \geq 1$,*

1. $\Pr(h(y) = k) = (1/2)^k$,
2. $\Pr(h(y) \geq k) = (1/2)^{k-1}$, *and*
3. $\Pr(h(y) \leq k) = 1 - (1/2)^k$.

Number the elements of S from 1 to n, and define $x_0 := -\infty$ and $x_{n+1} := \infty$, so that

$$-\infty = x_0 < x_1 < x_2 < \ldots < x_n < x_{n+1} = \infty.$$

Let m be the index such that $x_m \leq x < x_{m+1}$. Then $0 \leq m \leq n$.

Lemma 8.2 *The search paths to x_m and x are the same.*

According to this lemma, it suffices to analyze the expected time our algorithm needs when searching for element x_m. Let $path(x_m)$ denote the search path to x_m. Then the time to search for x_m is proportional to the number of nodes on $path(x_m)$.

If $m = 0$, then $path(x_m)$ is the path containing exactly all $-\infty$ nodes and, therefore, contains exactly h nodes. We know already that the expected value of h is less than or equal to $3 + \lceil \log n \rceil$. So we may assume that $m \geq 1$.

For any i, $0 \leq i \leq n$, and any $k \geq 1$, we define a random variable Y_{ik} whose value is equal to

$$Y_{ik} = \begin{cases} 1 & \text{if } x_i \in L_k \text{ and the occurrence of } x_i \text{ in } L_k \text{ lies on } path(x_m), \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 8.3 *The number of nodes on path*(x_m) *is equal to*

$$\sum_{i=0}^{m} \sum_{k=1}^{\infty} Y_{ik},$$

which is less than or equal to

$$h + \sum_{i=1}^{m} \sum_{k=1}^{\infty} Y_{ik}.$$

Lemma 8.4 *Let* $1 \leq i \leq m$ *and* $k \geq 1$. *Then* $Y_{ik} = 1$ *if and only if (i)* $h(x_i) \geq k$, *and (ii)* $h(x_\ell) \leq k$ *for all* ℓ, $i+1 \leq \ell \leq m$.

Lemma 8.4 implies that for $1 \leq i \leq m$ and $k \geq 1$,

$$\Pr(Y_{ik} = 1) \; = \; \Pr[h(x_i) \geq k \text{ and } h(x_\ell) \leq k \text{ for all } \ell,\, i+1 \leq \ell \leq m].$$

Since the random variables $h(\cdot)$ are independent, it follows that

$$\begin{aligned}
\Pr(Y_{ik} = 1) &= \Pr(h(x_i) \geq k) \cdot \prod_{\ell=i+1}^{m} \Pr(h(x_\ell) \leq k) \\
&= (1/2)^{k-1} \cdot \prod_{\ell=i+1}^{m} \left(1 - (1/2)^k\right) \\
&= (1/2)^{k-1} \cdot \left(1 - (1/2)^k\right)^{m-i}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
E(Y_{ik}) &= 0 \cdot \Pr(Y_{ik} = 0) + 1 \cdot \Pr(Y_{ik} = 1) \\
&= (1/2)^{k-1} \cdot \left(1 - (1/2)^k\right)^{m-i}.
\end{aligned}$$

The expected number of nodes on *path*(x_m) is equal to

$$E\left(\sum_{i=0}^{m} \sum_{k=1}^{\infty} Y_{ik}\right) \leq E\left(h + \sum_{i=1}^{m} \sum_{k=1}^{\infty} Y_{ik}\right).$$

In order to compute this expected value, we use the following result.

Lemma 8.5 *Let* $(Z_i)_{i \geq 1}$ *be an infinite sequence of random variables, all taking non-negative values. If* $\sum_{i=1}^{\infty} E(Z_i)$ *converges, then*

$$E\left(\sum_{i=1}^{\infty} Z_i\right) = \sum_{i=1}^{\infty} E(Z_i).$$

Proof. We have seen already that the linearity of expectation holds for sums consisting of a *deterministic* and *finite* number of terms. The infinite version follows from the monotone convergence theorem, whatever that is. It is a corollary to Theorem 16.7 in Billingsley, Probability and Measure. See also Section 7.2 in Loeve, Probability Theory 1, 4th edition. ■

Hence, the expected number of nodes on $path(x_m)$ is less than or equal to

$$E\left(h + \sum_{i=1}^{m} \sum_{k=1}^{\infty} Y_{ik}\right) = E(h) + \sum_{i=1}^{m} \sum_{k=1}^{\infty} E(Y_{ik}), \tag{10}$$

provided we can show that the summation on the right-hand side converges. This will follow from the computations that follow. We know already that

$$E(h) \leq 3 + \lceil \log n \rceil.$$

We will estimate the double summation on the right-hand side of (10). We have

$$\begin{aligned}
\sum_{i=1}^{m} \sum_{k=1}^{\infty} E(Y_{ik}) &= \sum_{i=1}^{m} \sum_{k=1}^{\infty} (1/2)^{k-1} \cdot \left(1 - (1/2)^k\right)^{m-i} \\
&= \sum_{k=1}^{\infty} (1/2)^{k-1} \sum_{i=1}^{m} \left(1 - (1/2)^k\right)^{m-i} \\
&= \sum_{k=1}^{\infty} (1/2)^{k-1} \sum_{\ell=0}^{m-1} \left(1 - (1/2)^k\right)^{\ell} \\
&= \sum_{k=1}^{\infty} (1/2)^{k-1} \frac{1 - \left(1 - (1/2)^k\right)^m}{1 - (1 - (1/2)^k)} \\
&= \sum_{k=1}^{\infty} 2\left[1 - \left(1 - (1/2)^k\right)^m\right].
\end{aligned} \tag{11}$$

Lemma 8.6 *For any real number z, $0 \leq z \leq 1/2$, we have* $\ln(1-z) \geq -2z$.

Exercise 8.7 Draw the graphs of the functions $\ln(1-z)$ and $-2z$, and "conclude" that the inequality in Lemma 8.6 holds. Then give a formal proof by computing the minimum, using elementary calculus, of the function $f(z) = 2z + \ln(1-z)$ for $0 \leq z \leq 1/2$.

Lemma 8.6 implies that for $m \geq 0$ and $0 \leq z \leq 1/2$,

$$(1-z)^m = e^{m \ln(1-z)} \geq e^{-2mz}. \tag{12}$$

Let $z = (1/2)^k$, where $k \geq 1$. Then $0 \leq z \leq 1/2$, and (12) becomes

$$\left(1 - (1/2)^k\right)^m \geq e^{-2m/2^k} = e^{-m2^{1-k}}. \tag{13}$$

Combining (11) and (13) implies that the double summation on the right-hand side of (10) is less than or equal to

$$2 \sum_{k=1}^{\infty} \left(1 - e^{-m2^{1-k}}\right).$$

We write this as $A + B$, where

$$A := 2 \sum_{k=1}^{\lceil \log n \rceil} \left(1 - e^{-m2^{1-k}}\right),$$

and

$$B := 2 \sum_{k=1+\lceil \log n \rceil}^{\infty} \left(1 - e^{-m2^{1-k}}\right).$$

Since each term in the summation of A is less than one, we have

$$A \leq 2\lceil \log n \rceil.$$

Lemma 8.8 *For any real number $z \geq 0$, we have* $1 - e^{-z} \leq z$.

Exercise 8.9 Prove Lemma 8.8.

Applying Lemma 8.8 with $z = m2^{1-k}$ gives

$$1 - e^{-m2^{1-k}} \leq m2^{1-k}.$$

Therefore,

$$\begin{aligned} B &\leq 2 \sum_{k=1+\lceil \log n \rceil}^{\infty} m2^{1-k} \\ &= 4m \sum_{k=1+\lceil \log n \rceil}^{\infty} (1/2)^k \\ &= 4m(1/2)^{\lceil \log n \rceil} \\ &\leq 4m \cdot \frac{1}{n} \\ &\leq 4. \end{aligned}$$

This proves that the expected number of nodes on the search path to element x_m is less than or equal to

$$E(h) + A + B \leq 3 + \lceil \log n \rceil + 2\lceil \log n \rceil + 4 = 3\lceil \log n \rceil + 7.$$

9 Further reading

Skip lists were invented by William Pugh in 1989. See [7, 8, 9]. Another randomized dictionary, based on binary search trees, was introduced in 1989 by Aragon and Seidel [10].

An introduction to probability theory can be found in the book by Cormen, Leiserson and Rivest [2]. The standard book on this topic is Feller [3]. Comprehensive overviews of randomized algorithms and data structures can be found in the books by Motwani and Raghavan [5], and by Mulmuley [6].

References

[1] M. Aigner and G. M. Ziegler. *Proofs from THE BOOK.* Springer-Verlag, Berlin, 1998.

[2] T.H. Cormen, C.E. Leiserson and R.L. Rivest. *Introduction to Algorithms.* MIT Press and McGraw-Hill, 1990.

[3] W. Feller. *An Introduction to Probability Theory and Its Applications*, Volume I, 3rd Edition. John Wiley & Sons, 1968.

[4] G.H. Gonnet, Journal of the ACM **28**, 1981, pp. 289–304.

[5] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, New York, 1995.

[6] K. Mulmuley. *Computational Geometry, an Introduction through Randomized Algorithms.* Prentice Hall, Englewood Cliffs, 1994.

[7] W. Pugh. *Skip lists: a probabilistic alternative to balanced trees.* Proc. WADS, Lecture Notes in Computer Science, Vol. 382, Springer-Verlag, Berlin, 1989. pp. 437–449.

[8] W. Pugh. *A skip list cookbook.* Technical Report, University of Maryland, 1989.

[9] W. Pugh. *Skip lists: a probabilistic alternative to balanced trees.* Communications of the ACM **33** (1990), pp. 668–676.

[10] R. Seidel and C.R. Aragon. *Randomized search trees.* Algorithmica **16** (1996), pp. 464–497.