COMP 2804 — Assignment 2 (Solutions)

Question 1: On the first page of your assignment, write your name, student number, and course number COMP 2804.

Question 2: A small computer network consists of 6 computers. Each computer is directly connected to 0 or more computers. Show that there are at least two computers in the network that are directly connected to the same number of other computers. Is the statement true if we have 10 computers instead of 6?

Solution: We show that the statement is true by using Pigeonhole principle. Since each computer can be directly connected to 0 or more computers, there are six choices of connections for each computer, i.e., 0, 1, 2, 3, 4 or 5 connections. But if one computer is connected to zero computer, then no computer can be connected to five others. Similarly if one computer is connected to five other computers, then no computer can be connected to zero others. This means that 0 and 5 can not be simultaneously the number of connections in a network of 6 computer. So, there are actually 5 choices and 6 computers. According to Pigeonhole principle at least two computers must have the same number of direct connections.

The statement is still true if we have 10 computers instead of 6; now we have 10 computers and 9 choices. Same reasoning holds for any network of n computers.

Question 3: Show that $a_n = n^2 + n + 1$ satisfies

$$\begin{cases} a_0 = 1 \\ a_k = a_{k-1} + 2k \quad \text{for } k > 0. \end{cases} \tag{1}$$

Solution: We prove by induction on n.

- Base case: If $n = 0$, then $a_0 = 1$ and $n^2 + n + 1 = 0 + 0 + 1 = 1$. It proves the base case holds.
- Induction hypothesis: Let $n \geq 1$, and assume that $a_{n-1} = (n-1)^2 + (n-1) + 1$ is true for $n - 1$.
- Induction step: We have to show that $a_n = n^2 + n + 1$ satisfies the given recurrence relation. Following the recurrence and applying the induction hypothesis, we get

  $$\begin{aligned} a_n &= a_{n-1} + 2n \\ &= (n-1)^2 + (n-1) + 1 + 2n \\ &= n^2 - 2n + 1 + n - 1 + 1 + 2n \\ &= n^2 + n + 1 \end{aligned}$$

Question 4: Consider the following recursive function defined for positive values of n

$$M(n) = \begin{cases} n-10 & \text{if } n > 100 \\ M(M(n+11)) & \text{if } n \leq 100 \end{cases} \tag{2}$$

Evaluate $M(120)$, $M(111)$, $M(97)$, $M(94)$. Do you observe something interesting for the value of $M(n)$ for $n \leq 100$?

Solution: Using the given recurrence function we evaluate the following terms:
$\mathbf{M(120)} = 120 - 10 = 110$

$\mathbf{M(111)} = 111 - 10 = 101$

$$\begin{aligned}\mathbf{M(97)} &= M(M(97+11)) = M(108) = M(108-10) = M(98)\\ &= M(M(98+11)) = M(109) = M(109-10) = M(99)\\ &= M(M(99+11)) = M(110) = M(110-10) = M(100)\\ &= M(M(100+11)) = M(111) = M(111-10) = M(101) = 101-10 = 91\end{aligned}$$

$$\begin{aligned}\mathbf{M(94)} &= M(M(94+11)) = M(105) = M(105-10) = M(95)\\ &= M(M(95+11)) = M(106) = M(106-10) = M(96)\\ &= M(M(96+11)) = M(107) = M(107-10) = M(97)\\ &= M(M(97+11)) = M(108) = M(108-10) = M(98)\\ &= M(M(98+11)) = M(109) = M(109-10) = M(99)\\ &= M(M(99+11)) = M(110) = M(110-10) = M(100)\\ &= M(M(100+11)) = M(111) = M(111-10) = M(101) = 101-10 = 91\end{aligned}$$

For all $n \leq 100$, $M(n) = 91$.

Question 5: Let d_n denote the number of ways that n-letters can be put into n-envelopes so that no letter goes into the correct envelope. Show that $d_1 = 0$, $d_2 = 1$, $d_3 = 2$, and in general for $n \geq 3$,

$$d_n = (n-1)(d_{n-1} + d_{n-2}).$$

Solution:
d_1=# of ways 1 letter can be put into 1 envelope so that it goes into the wrong envelope=0.

d_2= # of ways 2 letters can be put into 2 envelopes so that no letter goes into the correct envelope = (# of ways without restriction)-(# of ways all are in correct envelopes)-(# of ways 1 letter is in the correct envelope)= $2! - 1 - 0 = 1$.

d_3= # of ways 3 letters can be put into 3 envelopes so that no letter goes into the correct envelope = (# of ways without restriction)-(# of ways all are in correct envelopes)-(# of ways 1 letter is in the correct envelope)-(# of ways 2 letters is in the correct envelope)= $3! - 1 - \binom{3}{1} \times 1 - 0 = 2$.

For $n \geq 3$, suppose, $(n-1)$ letters $l_1, l_2, \cdots, l_{n-1}$ have been placed into $(n-1)$ wrong envelopes $e_1, e_2, \cdots, e_{n-1}$. Now we want to put l_n. There are $(n-1)$ possible wrong envelopes to put l_n into. Suppose it is put into the i'th envelope. There are two cases to consider:
Case 1: suppose l_i is put into e_n, then there are d_{n-2} ways to put all other letters into wrong envelopes.
Case 2: suppose l_i is not put into e_n, then there are d_{n-1} ways to put all other letters such that all these are in wrong envelopes.
So, when l_n is put into the e_i there are $(d_{n-1} + d_{n-2})$ ways to put all letters into wrong envelopes. But, there are total $(n-1)$ wrong envelopes for l_n. Finally we can see that,

$$d_n = (n-1)(d_{n-1} + d_{n-2}), \ \ n \geq 3$$

Question 6: Using induction for $n \geq 1$, show that d_n in the previous question can be expressed as

$$d_n = n!\Big(\frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} + \cdots + (-1)^n \frac{1}{n!}\Big).$$

Solution: The recurrence relation of the previous question can be rewritten as,

$$d_n - nd_{n-1} = (-1)[(d_{n-1} - (n-1)d_{n-2})]$$

Similarly,

$$d_{n-1} - (n-1)d_{n-2} = (-1)[(d_{n-2} - (n-2)d_{n-3})]$$

and so on till

$$d_3 - 3d_2 = (-1)[(d_2 - 2d_1)] = (-1)(1 - 2 \times 0) = (-1)$$

And we know that $d_2 = 1$ and $d_1 = 0$. Therefore, d_n can be redefined as

$$d_n - nd_{n-1} = (-1)^n, \ \ n \geq 1$$

Now,

$$\frac{d_n}{n!} - \frac{d_{n-1}}{(n-1)!} = \Big((-1)^n \frac{1}{n!}\Big),$$

Similarly,

$$\frac{d_{n-1}}{(n-1)!} - \frac{d_{n-2}}{(n-2)!} = \Big((-1)^{n-1} \frac{1}{(n-1)!}\Big),$$

and

$$\frac{d_2}{2!} - \frac{d_1}{1!} = \Big((-1)^2 \frac{1}{2!}\Big), \qquad \frac{d_1}{1!} - \frac{d_0}{0!} = \Big((-1)^1 \frac{1}{1!}\Big)$$

Adding all together, we obtain

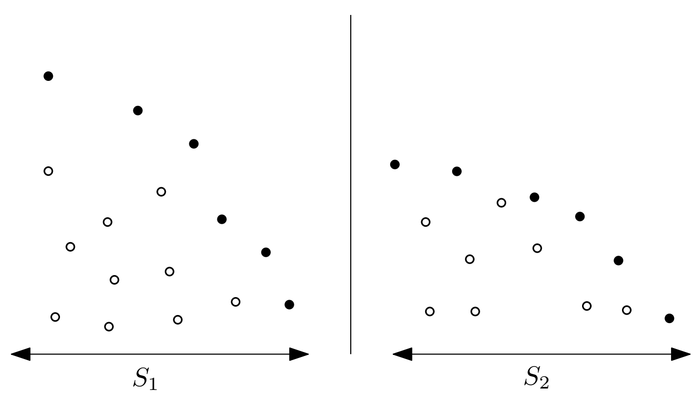
$$d_n = n!\Big(\frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} + \cdots + (-1)^n \frac{1}{n!}\Big).$$

Question 7: Exercise 4.36 from the Course Text Book by Michiel Smid.

Solution: We describe a recursive algorithm $MaxElem(S)$ that returns all maximal elements of S, in sorted order from left to right. Since the algorithm is recursive, we need a base case. We take the base case to be when $n = 1$. In this case, the input consists of one single point, which is obviously maximal. So we just return this single point.

Let $n \geq 2$. We split S in the middle and run the algorithm recursively on each part:

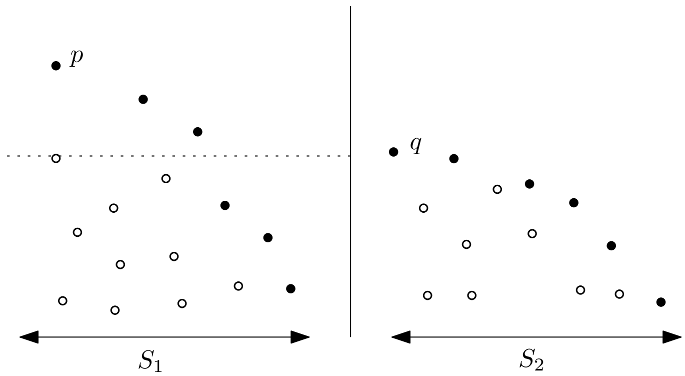
- Let $S1$ be the set of the first $n/2$ points in S. Run algorithm $MaxElem(S1)$ and let $M1$ be the output.
- $S2$ is the set of the last $n/2$ points in S. Run algorithm $MaxElem(S2)$ and let $M2$ be the output.



The merge step: observe the following,

- $M2$ contains all maximal elements in $S2$. Since $M2$ is to the right of $S1$, each point in $M2$ is maximal in the entire set S.
- $M1$ contains all maximal elements in $S1$. However, points in $M1$ may not be maximal in the entire set S. Let p be an arbitrary element in $M1$, and let q be the leftmost element in $M2$. Then p is maximal in S if and only if $p_y > q_y$.

From this, it follows that we can do the merge step in the following way:



- Take the leftmost point q in $M2$.
- Walk along $M1$ and add all points whose y-coordinate is larger than that of q to a list M.
- Add all points in $M2$ to the list M.
- Return the list M. This list contains all maximal elements in the entire set S.

The running time $T(n)$ is the sum of the following:

- The time to split S into two sets; this can be done in $O(n)$ time by scanning S.
- The time for the recursive call $MaxElem(S1)$; this takes $T(n/2)$ time.
- The time for the recursive call $MaxElem(S2)$; this takes $T(n/2)$ time.
- The time for the merge step; this can be done in $O(n)$ time.

Thus, we arrive at the recurrence $T(n) = O(n) + 2T(n/2) = O(n \log n)$, which is the same as the MergeSort recurrence.

Question 8: Give a recursive definition of the set of integers that are multiples of 3. Show that your definition indeed generates all the elements of the set $\{\ldots, -9, -6, -3, 0, 3, 6, 9, \ldots\}$.

Solution: We give the following recursive definition of the set S of integers that are multiples of 3:

1. $3 \in S$.
2. If $m, n \in S$, then $m + n \in S$.
3. If $m, n \in S$, then $m - n \in S$.

Now we show that the definition above indeed generates all the elements of the set $\{\ldots,-6,-3,0,3,6,\ldots\}$.

Let $S' = \{3n : n \in N\}$, which clearly is a set of integers that are multiples of 3. To prove that our definition of S also defines the set of integers that are multiples of 3, we need to prove that $S = S'$.

- We first prove by induction that $S \subseteq S'$.

The base element in S is equal to 3. Since $3 = 3 \cdot 1$, this base element is in the set S'. Consider 2 elements m and n in S and assume that they are also in the set T'. Thus we assume that $m = 3 \cdot a$ and $n = 3 \cdot b$ for some integers a and b. Following our definition we get two more integers $m+n$ and $m-n$ in S, and we have $m+n = 3a+3b = 3(a+b)$ and $m-n = 3a-3b = 3(a-b)$, both of these are multiples of 3. Therefore, they are in S' as well. Hence, $S \subseteq S'$.

- Next we prove by induction that $S' \subseteq S$.

The base case is when n = 1, $3 \cdot 1 \in S'$. By definition of S, the number 3 is an element of S. Induction step: Let $n \geq 1$ and assume that $3n$ is an element of S. Then, by the definition of S, both $3(n+1) = 3n+3$ and $3(n-1) = 3n-3$ are also elements of S. Therefore, $S' \subseteq S$.

Question 9: Show that

$$a_n = \frac{1-r^{n+1}}{1-r}, r \neq 1$$

satisfies the recurrence relation

$$\begin{cases} a_0 = 1 \\ a_k = a_{k-1} + r^k \quad \text{for } k > 0. \end{cases} \tag{3}$$

Solution: We prove by induction on n.

- Base case: If $n = 0$, then $a_0 = 1$ and $\frac{1-r^{0+1}}{1-r} = 1$, proving the base case holds.
- Induction hypothesis: Let $n \geq 1$, and assume that $a_{n-1} = \frac{1-r^n}{1-r}$, $r \neq 1$ is true for $n-1$.
- Induction step: We have to show that $a_n = \frac{1-r^{n+1}}{1-r}, r \neq 1$ satisfies the given recurrence relation.

Again following the recurrence and applying the induction hypothesis, we get
$a_n = a_{n-1} + r^n$
$= \frac{1-r^n}{1-r} + r^n$
$= \frac{1-r^n+r^n-r^{n+1}}{1-r}$

$= \frac{1-r^{n+1}}{1-r}$

Question 10: A binary tree is
- either one single node
- or a node whose left subtree is a binary tee and whose right subtree is a binary tree.

Show that any binary tree with n leaves has exactly $2n-1$ nodes.

Solution: The proof is by induction on the number of leaves n in the binary tree.

Base case: when $n = 1$. In this case, the tree consists of a single node (a leaf). Thus, the tree has exactly 1 node, which is equal to $2n - 1$.

Induction hypothesis: Let $n \geq 2$ and assume the claim is true for all binary trees with less than n leaves.

Now, let T be a binary tree with n leaves. Since $n \geq 2$, T has a left subtree and a right subtree. Let m denote the number of leaves in the left subtree. Then the right subtree has $n - m$ leaves.

Since $m < n$, the induction hypothesis implies that the left subtree has $2m - 1$ nodes. Since $n - m < n$, the induction hypothesis implies that the right subtree has $2(n - m) - 1$ nodes. The number of nodes in T is equal to the sum of the root, the number of nodes in the left subtree, and the number of nodes in the right subtree. Therefore, the number of nodes in T is equal to $1 + (2m - 1) + (2(n - m) - 1) = 2n - 1$.

Question 11: (Bonus Problem:) Assume you have a set $A = \{a_1, a_2, \ldots, a_{n+1}\}$ of $n + 1$ positive numbers such that $\sum_{i=1}^{n+1} a_i = 2n$. Prove or disprove that for any integer k, where $1 \leq k \leq 2n$, we can always find a subset $B \subseteq A$ such that the sum of elements of B equals k.

Solution: We prove this statement is true by induction on k. Note that the elements of A are not distinct. We assume that these elements are sorted in non-decreasing order.

Base case: When $k = 1$, $A = \{1, 1\}$ and $B = \{1\}$. Similarly, When $k = 2$, $A = \{1, 1\}$ and $B = \{1, 1\}$.

Induction hypothesis: We assume that for all values l, where $1 \leq l < k \leq 2n$ we can always find a subset $B \subseteq A$ such that the sum of elements of B equals l.

Now, we want to show that this is also true for any $k \leq 2n$.

Let $(a_1 + a_2 + \cdots + a_i) < k$, where i is minimum and

$(a_1 + a_2 + \cdots + a_i + a_{i+1}) > k$, for some $1 \le i \le n$.

Note that if $a_1 + a_2 + \cdots + a_i + a_{i+1} = k$, then we have nothing to prove.

We claim that $a_{i+1} < k$. We will prove this later.

Thus, we can write $a_1 + a_2 + \cdots + a_i + a_{i+1} = k + \beta$, where $\beta < a_{i+1} < k$. Since $\beta < k$, by our induction hypothesis there exists a subset S of numbers from $\{a_1 + a_2 + \cdots + a_i + a_{i+1}\}$ that add up to β (set $\beta = l$). Then the sum of the remaining numbers from $\{a_1 + a_2 + \cdots + a_i + a_{i+1}\} \setminus S$ equals to k and we are done.

Proof of claim that $a_{i+1} < k$.
Proof: Observe that

$$i < k$$

or

$$k \ge i + 1 \tag{4}$$

Consider once again the numbers:

$$\begin{array}{cccccccc}
a_1 & a_2 & a_3 & \cdots & a_i & a_{i+1} & \cdots & a_n & a_{n+1} \\
\| & \| & \| & \cdots & \| & \| & \cdots & \| & \| \\
\ge 1 & \ge 1 & \ge 1 & \cdots & \ge 1 & \ge k+1 & \cdots & \ge k+1 & \ge k+1
\end{array}$$

Let us take the least possible values for these numbers, and we will show that their sum exceeds $2n$.
That is, consider the sum
$(a_1 + a_2 + \cdots + a_i) + (a_{i+1} + \cdots + a_n + a_{n+1})$
$\ge (1 + 1 + 1 + \cdots + 1) + (k + 1 + k + 1 + \cdots + k + 1)$
$= i + (n + 1 - i)(k + 1)$

Now, we claim that $i + (n + 1 - i)(k + 1) > 2n$.
Proof. Since $k \ge i + 1$ by equation (4),
if

$$i + (n + 1 - i)(i + 1 + 1) \ge 2n$$

or

$$i + (n + 1 - i)(i + 2) \ge 2n$$

or

$$i + ni + i - i^2 + 2n + 2 - 2i \ge 2n$$

or

$$ni + 2 - i^2 \ge 0$$

Since $n \ge i$, the above equation holds.