Lecture 2: PRODUCT RULE

> Assume a procedure consists of performing a sequence of m tasks in order. Let there be N_1 ways to do first task,
> N_2 ways to do second task,
> $\vdots$
> N_m ways to do the m^{th} task.
>
> Then there are $N_1 N_2 N_3 \cdots N_m$ ways to do the entire procedure.

Examples:

1. # bit strings of length n.
2. # functions $f: A \to B$ where $|A| = m$, $|B| = n$.
3. # one-to-one functions $f: A \to B$
4. # ways to place m-books on n-book shelves

WARM-UP

CTC Center: seats are numbered by

three parameters [Section #, Row #, Seat #]
76 16 18

What is the total # of seats?

CTC Problem:

View the above as a "procedure" of writing down a string of three characters: first being Section #

second being Row #

third being Seat #

seats ≡ How many ways to perform this procedure?

THIS Procedure consists of three tasks:

1st Task: WRITE DOWN SECTION #

2nd Task: WRITE DOWN ROW #

3rd Task: WRITE DOWN SEAT #

Let us analyze # ways to do each task.

<u>Observe</u>: $m=3$; $N_1 = 76$ as there are 76 ways of performing first task.

$N_2 = 16$, as in 2nd task we can pick any row irrespective of which Section we choose in 1st task.

$N_3 = 18$, as in 3rd task we can pick any seat irrespective of which Section/Row we picked in the first two tasks.

Thus # seats ≡ # ways to execute the procedure

$$\equiv N_1 N_2 N_3$$

$$\equiv 76 * 16 * 18$$

$$\equiv 21888.$$

Examples

1. How many bit strings of length n are there?

e.g when $n=3$, the bit strings are
$$\begin{cases} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{cases}$$

We apply product rule.

→ Procedure ≡ write down a bit string of length n.

→ Tasks ≡ For $i := 1, 2, \ldots, n$,

i-th task is to write down a bit.

Observe: There are two ways to do the i^{th} task, irrespective of how we did the first $i-1$ tasks.

$$\Rightarrow N_i = 2$$

This is true for each value of $i = 1, 2, \ldots, n$.

ways to perform procedure

$$= N_1 N_2 N_3 \cdots N_n = \underbrace{2 * 2 * \cdots * 2}_{n\text{-times}} \equiv 2^n.$$

2. How many numbers between 100 and 1000 have three distinct odd digits?

We want to generate numbers of the form

357, 371, 159, ...

→ Procedure : Generate three digit numbers with distinct odd digits

→ Tasks : For $i := 1$ to 3, generate the i^{th} digit.

Task-1 : Choose the first digit among $\{1,3,5,7,9\}$.

$N_1 = 5$

Task-2 : Choose the second digit among $\{1,3,5,7,9\}$ but distinct from first one.

$N_2 = 4$

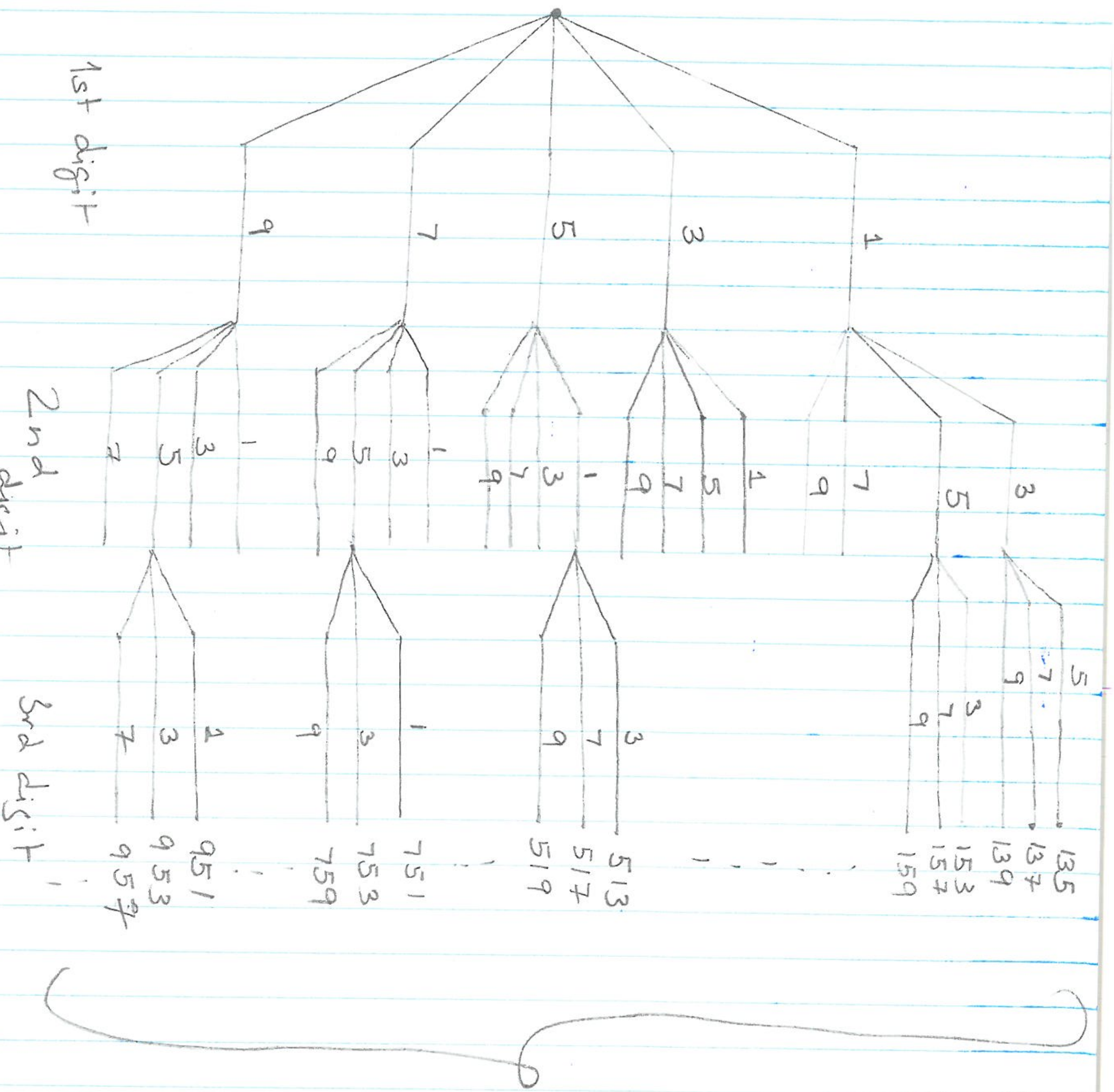
Task-3 : Choose the last digit among $\{1,3,5,7,9\}$ but distinct from first & second.

$N_3 = 3$.

Thus : $\#\text{Numbers} \equiv N_1 N_2 N_3 = 5 * 4 * 3$

$= 60$

Visualization

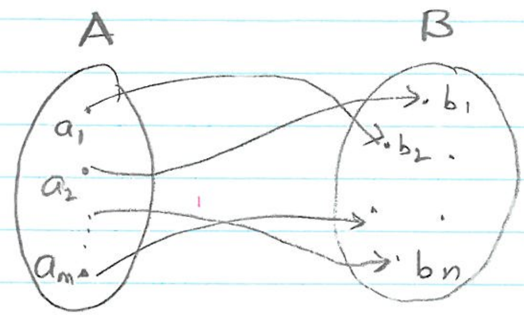


$5 * 4 * 3$

$= 60$ ways.

2. Let $f : A \to B$, $|A| = m \geq 1$, $|B| = n \geq 1$. $A = \{a_1, a_2, \ldots, a_m\}$, $B = \{b_1, b_2, \ldots, b_n\}$.

f :



Any function $f : A \to B$ is completely specified by $(f(a_1), f(a_2), \ldots, f(a_m))$; where $f(a_i) \in B \;\forall i \in \{1..m\}$

Q: How many functions $f : A \to B$ are there?

We will apply PRODUCT RULE.

→ Procedure: Specify values $f(a_1), f(a_2), \ldots, f(a_m)$

→ Tasks: For $i := 1, 2, \ldots, m$, i-th task is to specify the value of $f(a_i)$.

<u>Observe</u>: For each $i \in \{1, 2, \ldots, m\}$, $f(a_i)$ can be any of the n elements of B.

⇒ There are $N_i = n$ ways to perform ith-task irrespective of how the first $i-1$ tasks are executed.

Thus,

by product rule,

ways to perform the procedure

$\equiv N_1 * N_2 * \cdots * N_m$

$\equiv \underbrace{n * n * \cdots * n}_{m \text{ times}}$

$\equiv n^m$

<u>Theorem</u>: Let $m \geq 1$, $n \geq 1$ be integers.

Let A be a set of m-elements.

Let B be a set of n-elements.

There are n^m functions $f: A \to B$.

---

3 Counting one-to-one function $f: A \to B$.

$|A| = m \geq 1$, $|B| = n \geq 1$.

A function $f: A \to B$ is one-to-one if for any $1 \leq i \neq j \leq m$, $f(a_i) \neq f(a_j)$.

Q: How many one-to-one function $f: A \to B$ are there.

Case 1; $|A| > |B|$ then $f: A \to B$ cannot be one-to-one.

Case 2: $|A| \leq |B|$.

We will use the Product Rule.

→ PROCEDURE: Specify values $f(a_1), f(a_2), \dots, f(a_m)$

→ Tasks: for $i := 1, 2, \dots, m$, the i-th task is to specify the value of $f(a_i)$.

→ Let us analyze # of ways of doing these tasks.

Task-1; $f(a_1)$ can be any of n-values of B.

Thus, $N_1 = n$.

Task-2; $f(a_2)$ can be any of values of B, except what was assigned to $f(a_1)$.

Thus, $N_2 = n - 1$

i.e, there are $n-1$ ways to do task 2 irrespective of how we did task 1.

Task-3 : $f(a_3)$ can be any of the values in B, except what was assigned to $f(a_1)$ and $f(a_2)$

Regardless of how we do the first two tasks, there are

$$N_3 = n-2$$ ways to do 3rd-task.

⋮

Task-i : $f(a_i)$ can be any of the values in B, except what was assigned to the first $i-1$ tasks.

There are $N_i = n-(i-1) = n-i+1$ ways to do the i^{th} task.

Thus, by product rule

\# ways to execute procedure

$$= N_1 N_2 \cdots N_m$$

$$= n(n-1)(n-2)\cdots(n-m+1)$$

$$= n(n-1)(n-2)\cdots(n-m+1)\frac{(n-m)(n-m-1)\cdots 1}{(n-m)(n-m-1)\cdots 2 \cdot 1}$$

$$= \frac{n!}{(n-m)!}$$

Theorem: $f: A \to B$, $|A| = m \geq 1$, $|B| = n \geq 1$

(a) If $|A| > |B|$ then there are 0 one-to-one function $f: A \to B$.

(b) If $|A| \leq |B|$ then there are $\dfrac{n!}{(n-m)!}$ functions $f: A \to B$

4. Placing m-Books on n-Shelves

Let $\mathcal{B} = \{B_1, B_2, ---, B_m\}$ be a collection of m-books.

Let $\mathcal{S} = \{S_1, S_2, ---, S_n\}$ be n-book shelves.

We need to place all books into the shelves, where by placement we mean

- for each book, specify which shelf it will go, and
- specify the left-to-right order of books in a shelf.

Assume: Any shelf has enough capacity to take all the books.

Q: How many ways are there to place books on shelves?

We will apply the Product Rule.

→ Procedure : Place m books on n-shelves

→ Tasks : For $i = 1, 2, \cdots, m$, place B_i on to a shelf.

Note : B_i can be placed

(a) Leftmost on any shelf
or (b) to the right of a book among $B_1, B_2, \cdots, B_{i-1}$

Let us analyze # ways to do each task :

Task-1 : Place B_1 on to a shelf.

Since all shelves are empty to begin with, B_1 can be placed in $N_1 = n$ ways.

Task-2 : Place B_2 on to a shelf, given that B_1 is already placed

We have following two possibilities for placing B_2.

- We place B_2 on far-left on any shelf

or - We place B_2 immediately to right of B_1.

Thus $N_2 = n + 1$.

i^{th}-Task: Place B_i on to a shelf given that first $i-1$ books have already been placed.

Again, there are two possibilities

- Place B_i on far left of any of the shelves.

- Place B_i immediately to the right of one of $B_1, B_2, \ldots, B_{i-1}$.

Thus $N_i = n + i - 1$.

i.e, there $n+i-1$ ways to place B_i.

By the product rule,

ways to execute procedure

$$\equiv N_1 N_2 \cdots N_m$$

$$\equiv n \cdot (n+1)(n+2) \cdots (n+m-1)$$

$$\equiv \frac{1 \cdot 2 \cdot 3 \cdot \cdots (n-1)}{1 \cdot 2 \cdot 3 \cdots (n-1)} n(n+1)(n+2) \cdots (n+m-1)$$

$$\equiv \frac{(n+m-1)!}{(n-1)!}$$

∎