Design and Analysis of Algorithms

20 November, 2024

Q. Linear probing and quadratic probing এ probe number এর difference কেমন?

→ Linear probing এ বেশি-

- Unsuccessful search এ একটি space occupied পাওয়ার probability $\frac{n}{m}$

  পরেরটি occupied পাওয়ার probability: $\frac{n-1}{m-1}$

  তার পরে : $\frac{n-2}{m-2}$

∴ Total:

$$\frac{n}{m} \cdot \frac{n-1}{m-1} \cdot \frac{n-2}{m-2} \cdots \frac{n-i+1}{m-i+1} \leq \left(\frac{n}{m}\right)^{i} - 2$$

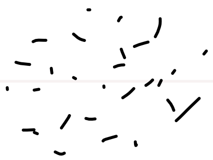
$$= \frac{1}{1-\alpha} \quad \left(\alpha = \frac{n}{m}\right)$$

unsuccessful search

→ এই ক্ষেত্রে used random variable হলো- discrete random variable.

→ successful search এ টি indicator random variable

Convex Hull



Some points are given. We have to determine the smallest possible area that cover all the points.

1) Graham Scan → finding the boundary of polynomial.

→ Algorithm শেষ হওয়ার পরে এই pointগুলো stack এ থাকবে, those are the boundary points of the required polygon.

• For collinear points, we will consider the farthest since I want to find the boundary without losing anything.

Incourse syllabus: Trie, suffix Tree, Hashing, NP

Exam এ proof ও আসতে পারে, simulation ও আসতে পারে

· RK took this course of 26.

25/11/2024

NUMBER THEORY (Chapter 31)

- Chinese Remainder theorem ← review on my own
- GCD (Euclidean Algorithm)

$$GCD(a, b) = gcd(b, a \bmod b)$$

terminal condition:

if $b == 0$ return a

- Extended Euclidean:

$$d = ax + by$$

Extended_Ec (a, b)

if $(b == 0)$ return $(a, 1, 0)$ ← gcd, x, y

else $(d', x', y') = extended\_ec(b, a \bmod b)$

$(d, x, y) = (d', y', x' - \lfloor \frac{a}{b} \rfloor y')$ ← floor

return (d, x, y)

Simulate: 99, 78

a	b	d	x	y
99	78	3	-11	14
78	21	3	3	-11
21	15	3	-2	3
15	6	3	1	-2
6	3	3	0	1

return	3	0	3	1	0

condition met

এই column হলো a mod b

এই column হলো $y = x' - \lfloor \frac{a}{b} \rfloor y'$

নিচের row   own row

এখানে একটি $\lfloor \frac{a}{b} \rfloor$ column maintain করা easier হবে

Q. what is the application of extended euclidean algorithm?

- <u>Euler Totient Function</u>

$\varphi(n)$ = the number of integers between 1 and n (inclusive) which are co-prime to n.

1. If n is prime, then $\varphi(n) = n-1$
2. If n is semi-prime (can be expressed as product of two primes p and q)

$$\varphi(n) = (p-1)(q-1)$$

3. If n is composite,

$$\varphi(n) = n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\dots$$

যতগুলো prime এর product is n

Fermat's Little Theorem

$$a^{p-1} \equiv 1 \pmod{p} \quad \text{(congruency)}$$

basically $a^{p-1} \bmod p = 1$ (equality)

$p =$ a prime

If p is not prime but a, n are co-prime

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

(Euler) (Generalized)

RSA

key → public (সবাই জানে)
key → private

when $A \xrightarrow{\text{message}} B$

- A will encrypt the message with B's public key. It becomes a cipher text C.
- B can decrypt the text C with its own private key

message কে public দিয়ে encrypt করলে private দিয়ে decrypt করতে পারবো, again private key দিয়ে encrypt করলে public key দিয়ে decrypt করতে পারবো। It is commutative.

<u>Challenge</u>: Key Generation

Let, $n = p \times q \rightarrow$ prime

$\varphi(n) = (p-1)(q-1)$

$1 < e < \varphi(n)$ and $\underbrace{\gcd(\varphi(n), e) = 1}_{\text{relative prime}}$

$d \equiv e^{-1} \pmod{\varphi(n)}$

$\therefore de \equiv 1 \pmod{\varphi(n)}$

multiplicative inverse of e

> Public key : $\{e, n\}$
>
> Private key : $\{d, n\}$

আমার কাছে $\varphi(n)$ আছে মানে আমার কাছে সব আছে। But n কে p, q তে factorize করাই challenging.

p, q হলো 1024 bit এর।

Simulate: $p = 3,\ q = 11$

$\phi(n) = 20$

Let, $e = 7$ (satisfies the two conditions)

Now, we have to find d.

$de \bmod 20 = 1$

Let, $d = 3$

$\therefore de = 21$

$\therefore 21 \bmod 20 = 1$

Encryption

$c = M^e \bmod n$

Let, M (message) = 100

$M^e = 100^7$

$= (10^2)^7$

$= 10^{14}$

$n = p \times q$

$= 33$

Now, $10^{14} \bmod 33 = 1 \longrightarrow (100 \bmod 33)^7$

$= (1)^7$

$= 1$

Decryption

$M = c^d \bmod n$

$= 1^3 \bmod 33$

$= 1 \bmod 33$

$= 1$

$\downarrow$

same m পাবোনা, because this will always give a value less than n. So, M must be less than n.

Restriction on M

– This method works due to modular arithmetic.

27 November, 2024

Prove: Any integer can be expressed as a product of primes.

→ induction দিয়ে করলে ভালো হবে।

Basis: 2

do on my own

→ Prove: This factorization will be unique

CRS: Complete Residue System

mod 8

$\longrightarrow \{0, 1, 2, 3, 4, 5, 6, 7\}$

RRS: Reduced Residue System (আগেরগুলো নিব যারা 8 এর সাথে co-prime)

$\{1, 3, 5, 7\}$

$\therefore \varphi(8) = 4$

We want to prove Euler's theorem:

$$a^{\varphi(n)} \equiv 1 \bmod (n)$$

Here, let, $a = 3$ [random number which is coprime with n]

$aa_i \rightarrow \{3, 9, 15, 21\}$

$aa_i \not\equiv aa_j \bmod n \quad (i \neq j)$

This is a property of reduced residue system.

Here,

$a \equiv b \bmod (n)$

implies n divides $a-b \rightarrow n \mid a-b$
(notation)

Case 1: $n = 1$

$\varphi(1) = 1$

$a^{\varphi(n)} = a^1 \equiv 1 \bmod (1)$

because $1 \mid a-1$

Case 2: $n > 1$

Let, $a_1, a_2, \ldots, a_{\varphi(n)}$: RRS mod n

Then $\gcd(a_i, n) = 1$, $a_i \not\equiv a_j$
$i \neq j$

??

We want to show a_{ai} is also RRS mod n.

WHAT THE FUCK??

$aa_1, aa_2, \ldots, aa_{\varphi(n)}$ is RRS

Then we can say

$$(aa_1)(aa_2)\cdots(aa_{\varphi(n)}) \equiv a_1 a_2 \cdots a_{\varphi(n)} \bmod n$$

[Congruent]

or, $$a^{\varphi(n)}(a_1 a_2 \ldots a_{\varphi(n)}) \equiv (a_1 a_2 \ldots a_{\varphi(n)}) \bmod n$$

$$\therefore a^{\varphi(n)} \equiv 1 \bmod n$$

RSA

M, e, d

$C = M^e \bmod n$

Now,

$$M = C^d \bmod n$$

$$= (M^e \bmod n)^d \bmod n$$

$$= M^{ed} \bmod n \quad \text{(modular arithmetic)}$$

$$= M \bmod n \rightarrow$$ This is where we want to reach

we know,

$$ed \equiv 1 \bmod \varphi(n)$$

$$\equiv 1 \bmod (p-1)(q-1)$$

$$\therefore ed = 1 + k(p-1)(q-1)$$

Now, from the pov of p,

$$M^{ed} \equiv M^{1+k(p-1)(q-1)} \bmod p$$

$$\equiv M \cdot M^{k(p-1)(q-1)} \bmod p$$

$$\equiv M \cdot \left(M^{p-1}\right)^{k(q-1)} \bmod p$$

$$\equiv M \left((M \bmod p)^{p-1}\right)^{k(q-1)} \bmod p$$

$$\equiv M \cdot (1)^{k(q-1)} \bmod p \quad \text{[Using Fermat's little theorem]}$$

$$\equiv M \bmod p$$

এখানে we are assuming m is not a multiple of p

যদি multiple হয় then we have to do an extra step:

$$m \equiv 0 \pmod p$$

Similarly, $M^{ed} \equiv M \bmod q$

$$\therefore M^{ed} \equiv M \bmod n$$

(এইটা পড়ার সময় modular arithmetic এর জিনিসগুলা পড়ে নিব, পাশে রাখবো তাহলে পড়া easier হবে)

আজকের class reading assignment → 23 Dec

Exam এ reduction part টা clear করে লিখতে হবে।

NP most important

Hashing এর proofs পড়ব

Trie and Suffix Tree এর input এর ক্ষেত্রে difference কী? (Trie এর ক্ষেত্রে separator আছে, suffix tree এ কোনো blank space নাই। We have to focus on these conceptual things)

— what are the search approaches, কী কী search করি text এ, trie কে ব্যবহার কেমন করব?

— Occurence এর location গুলো কোন কোন জায়গায়। একবার পেলেই শেষ না, we need all possible occurences.

23 December, 2024

- Correctness of RSA বুঝার জন্য congruence, Fermat's law নিয়ে পড়াশোনা করতে হবে।

- Proof of Euler theorem

What is CRS → complete Residue System
What is RRS → Reduced Residue System

ক $a^{\phi(n)} \equiv 1 \pmod{n}$

- Proof for $n=1$ and for $n>1$.

(i) $a \equiv a \mod(m)$

(ii) $a \equiv b \pmod{m}$

$\Rightarrow b \equiv a \pmod{m}$

(iii) $a \equiv b \pmod{m}$

& $b \equiv c \pmod{m}$

$\Downarrow$

$a \equiv c \pmod{m}$

(iv) $a \equiv b \pmod{m}$

and $c \equiv d \pmod{m}$

$$\Downarrow$$

$$ax + cy \equiv (bx + dy) \bmod m$$

$$a + c = (b + d) \bmod m$$

Prove $a \equiv b \pmod{m}$ iff a and b have the same remainder wrt m.

$$a = q_1 m + r_1$$
$$b = q_2 m + r_2$$

$$\Rightarrow a - b = (q_1 - q_2)m + r_1 - r_2$$

Since $m \mid (a-b)$, $r_1 - r_2$ must be 0

$$\therefore r_1 - r_2 = 0$$

$\therefore r_1 = r_2$, both remainders are same.

opposite side ও করতে হবে।

CHINESE REMAINDER THEOREM

Let, $m_1, m_2, \ldots m_n$ be relatively prime numbers. such that $\gcd(m_i, m_j) = 1$; $i \neq j$ then the linear congruence

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

$$\vdots$$

$$x = a_n \pmod{m_n}$$

has a simultaneous solution which is unique modulo$(m_1, m_2, \ldots m_n)$

- Proof আসবে।

$$m = m_1 m_2 \ldots m_n$$

$$M_1 = \frac{m}{m_1} = m_2 \ldots m_n$$

$$M_2 = \frac{m}{m_2} = m_1 m_3 \ldots m_n$$

$$M_n = \frac{m}{m_n} = m_1 m_2 \ldots m_{n-1}$$

$$\gcd(M_1, m_1) = 1$$

$$\gcd(M_k, m_k) = 1 \quad \forall\ k = 1 \ldots n$$

$M_k x_k \equiv 1 \pmod{m_k}$ (any random x_r আগের defined হয় না)

• $x^* = a_1 M_1 x_1 + a_2 M_2 x_2 + \cdots + a_n M_n x_n$

$m_k \mid M_i$ $(i \neq k)$ $M_i \equiv 0 \pmod{m_k}$

$= a_i m_i x$

এই জায়গায় ২টা term 0 হবে, বাকি কোনটা,

$a_k m_k x_k = 0$

Proof এর ২টা part → solution বের করতে হবে, then দেখাতে হবে solution টা unique.

30·12·2024

Q. Real life application of Chinese Remainder Theorem is what?

Concrete example → next class

Q- What is simultaneous solution?

Q. Euler's theorem এ $m=1$ case আলাদা করে দেখাতে হবে।

CRT তে $x = a_i M_i y_i$ কেন? Simultaneous solution দিয়ে দেখাতে হবে,

CRT চাইলে CLRS এর পাশাপাশি we can read Discrete Math book (Stewart)

BACKTRACKING

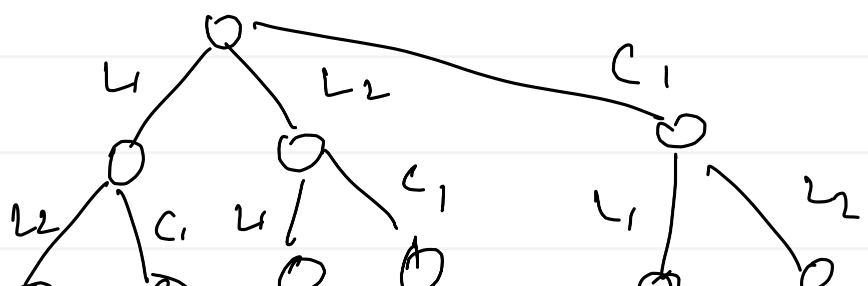
- যেসব problem এ একাধিক solution exist করে,
- Brute force.
- Backtracking vs DP

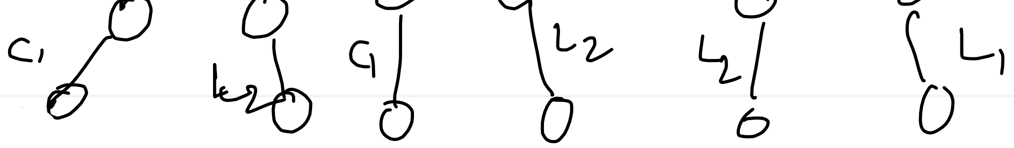
CSP : Constant satisfaction Problem

– যেখানে আমাদের কিছু শর্ত পূরণ করতে হবে,

- Solution is represented as State Space Tree (SST)

Let, L_1, L_2, C_1





leaf nodes are giving the solutions.

Let, we place a restriction that C_1 cannot be in the middle.

$\therefore 6 - 2 = 4$

Q. How to design the bounding function?

- N-Queen Problems

NXN size chessboard এ N টি queen বসানো।

Queen can attack row, column & diagonal wise.

∴ একই row, column, diagonal এ multiple queen থাকতে পারবেনা।

Let, N = 4

	Q_1		
			Q_2
Q_3			
		Q_4	

Q_1 place করে দেখব Q_2 place করা যাচ্ছে কিনা। and so on. place না করা গেলে আবার Q_1 এর জায়গা change করবো।

SUM OF SUBSETS / SUBSET-SUM PROBLEM

- we have a set of number. We want to find subsets which have a certain sum.

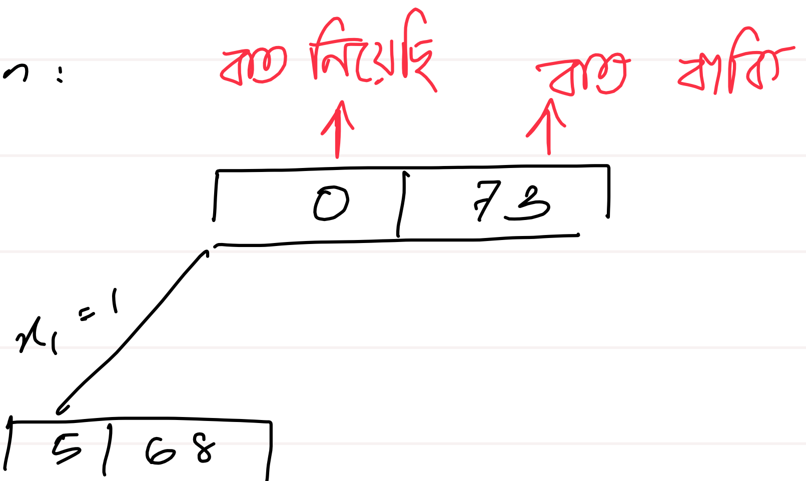
$w = \{5, 10, 12, 13, 15, 18\}$

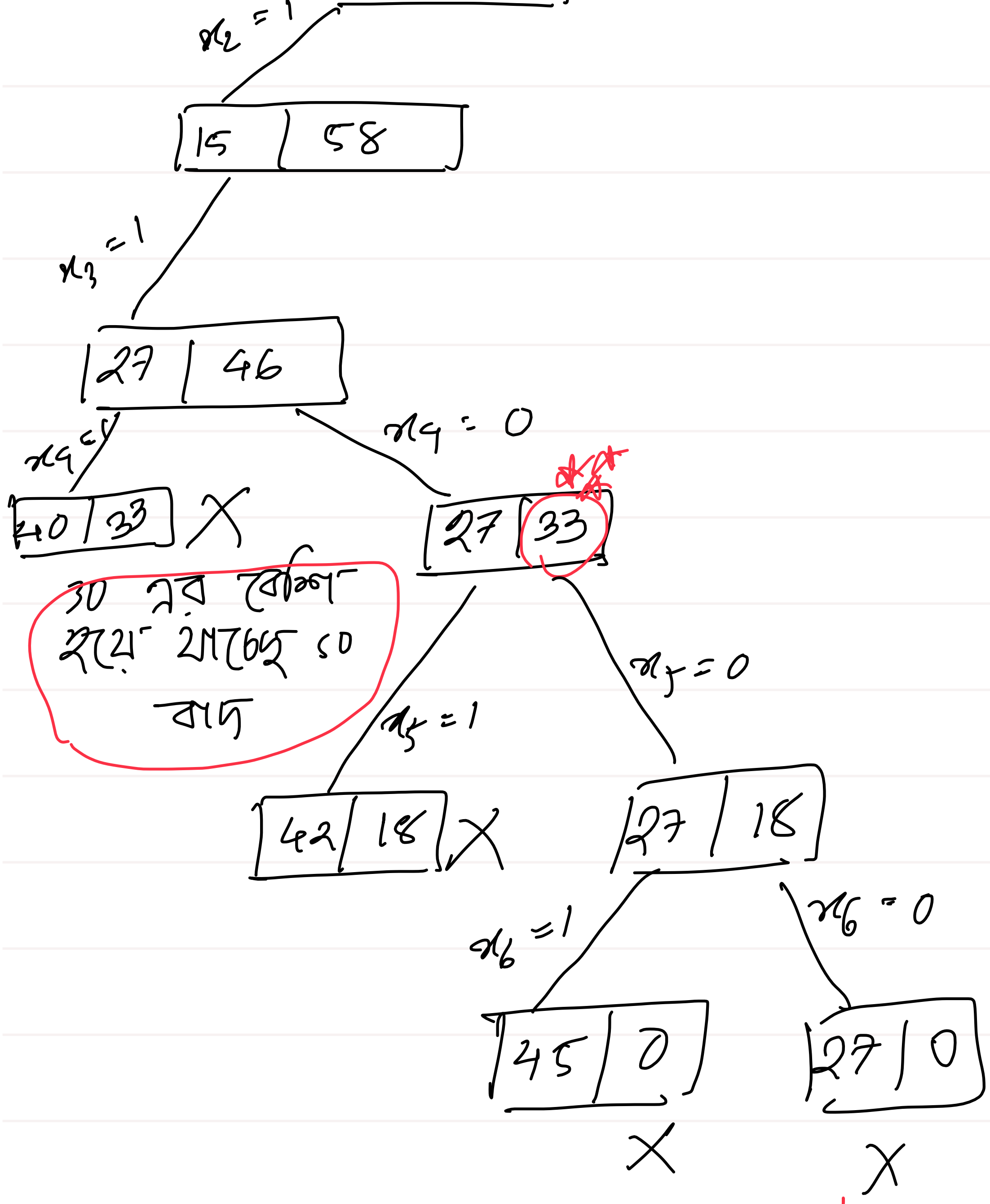
sum = 30

1	2	3	4	5	6
1	1	0	0	1	0

যেই element include করব সেটা 1.

The solution:





taken weight + remaining weight 30 এর চেয়ে কম so বাদ

∴ Terminating condition:

$$\sum_{i=1}^{k} w_i + \sum_{i=k+1}^{n} w_i < m$$

Explore করার condition:

$$\sum_{i=1}^{k} w_i + w_{k+1} \leq m$$

একটি solution পর্যন্ত leaf টা complete করে আনতে হবে।

<u>1/1/2025</u>

Subset sum problem এ set টা sorted থাকা, না থাকায় কী কোনো difference আছে?

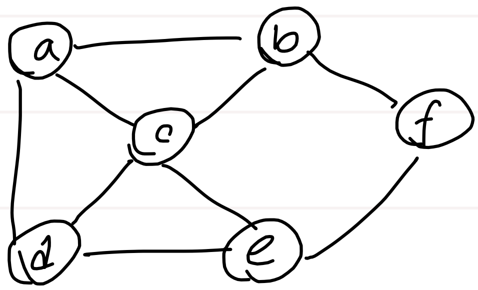
→ sorted না থাকলে problem নাই, but থাকলে কিছু advantage আছে। Easily বুঝতে পারবো কখন we should stop checking the next elements

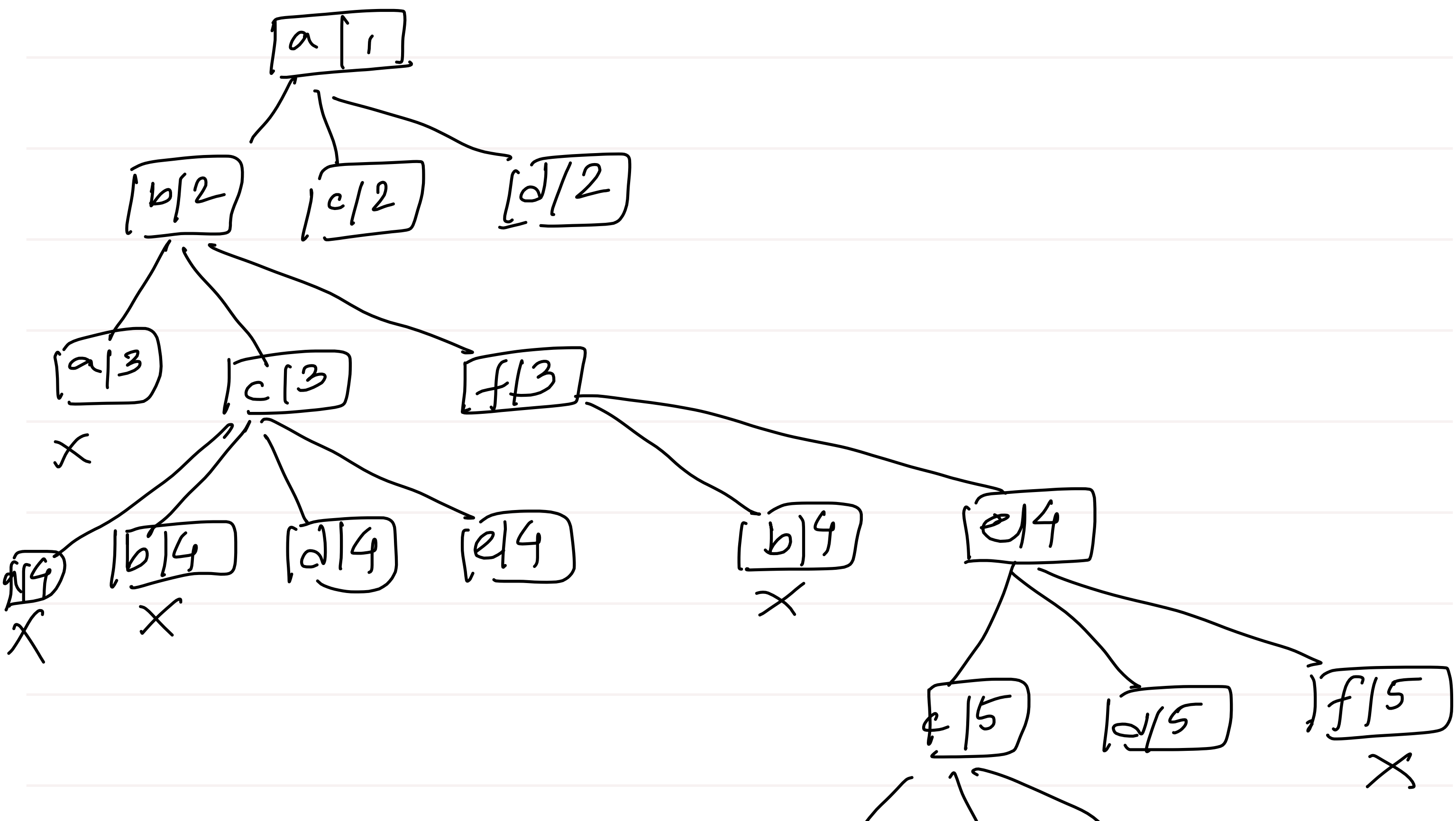
FEATURES OF BACKTRACKING

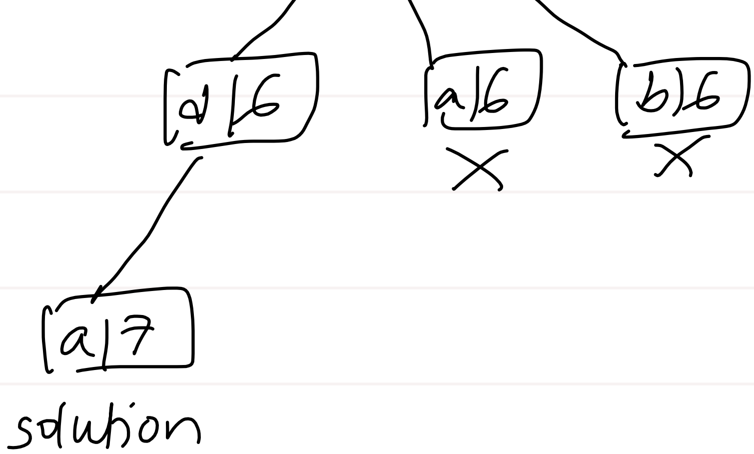
Backtracking → we find a feasible solution, and branch and bound approach → we find the optimal solution.

HAMILTONIAN CIRCUIT

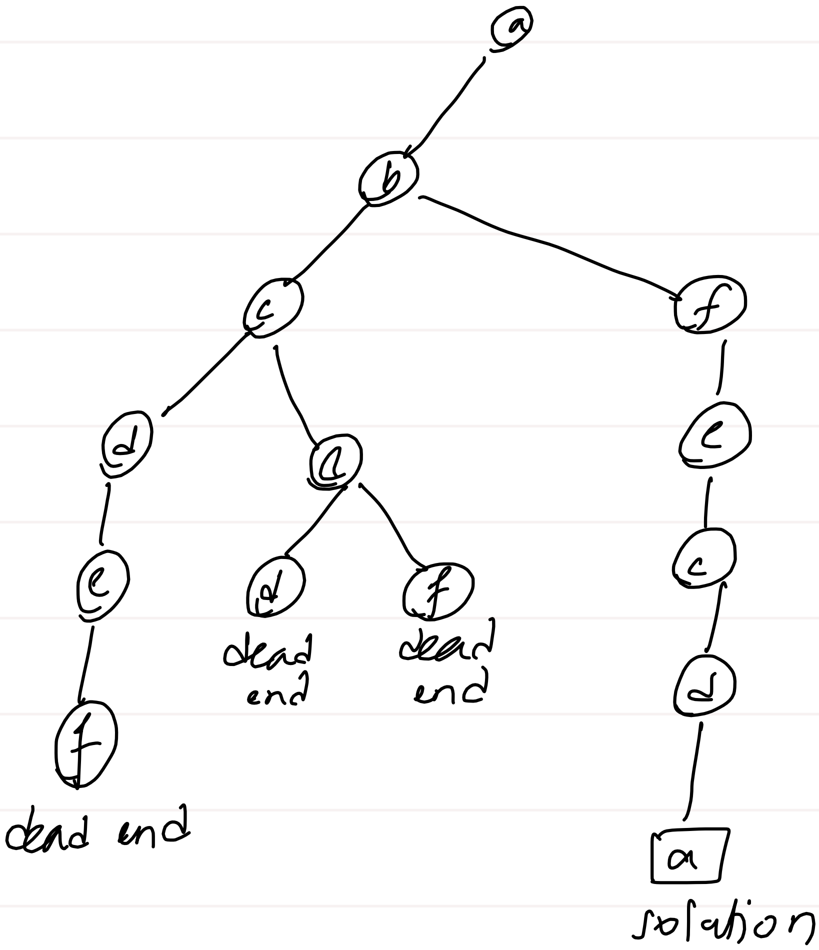
- visit all the nodes and create a cycle.







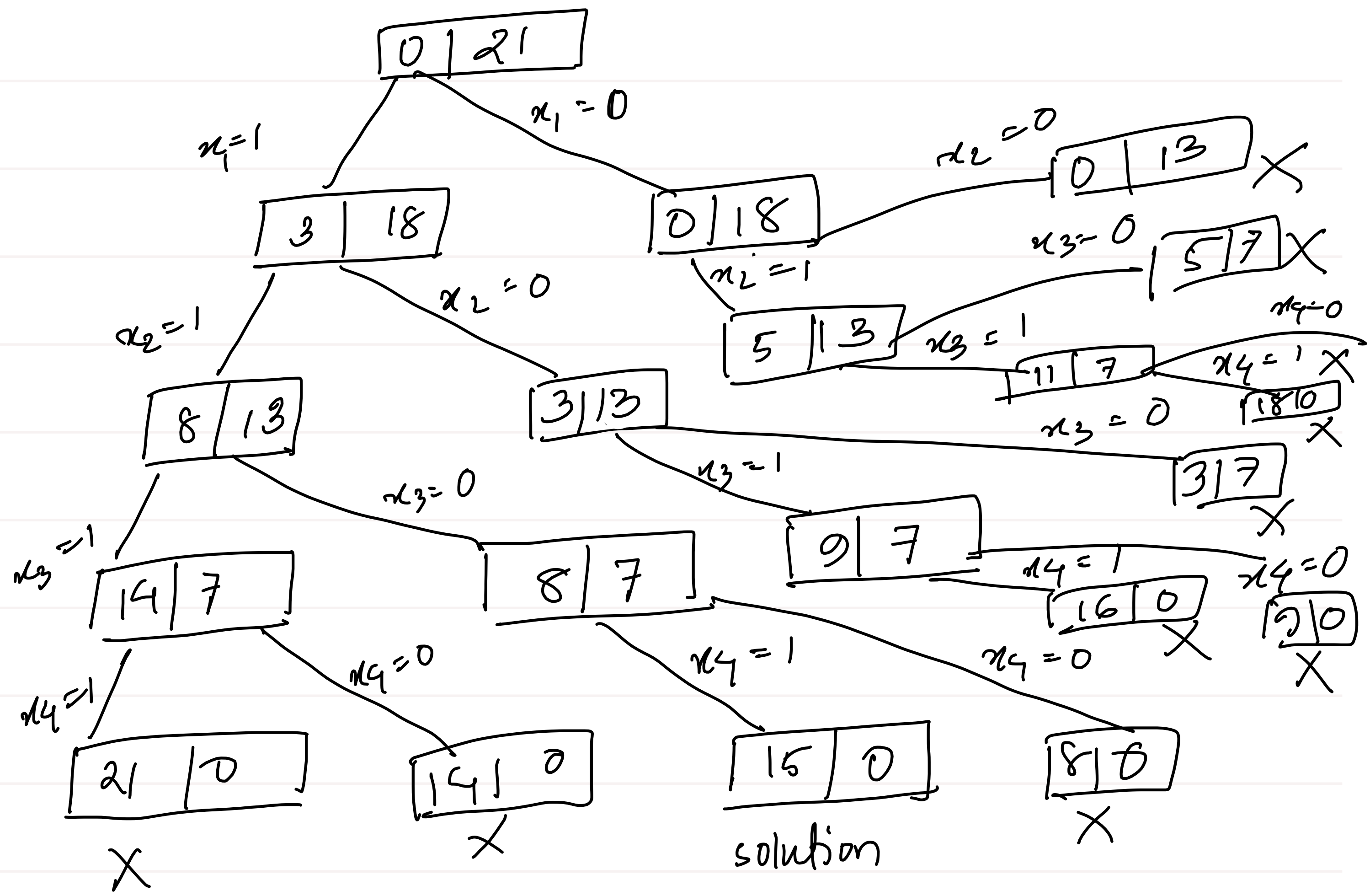
এভাবে করবনা। BFS approach না, DFS approach এ আঁকবো।



Q. Draw the state space tree of this subset sum problem:

$S = \{3, 5, 6, 7\}$

$d = 15$



Define the bounding functions formally

(B) Branch and bound এ we want the optimal solution, lets minimal lengths minimum cost, etc. So, backtracking এর চেয়ে বাড়তি কাজ করতে হবে।

Those two additional items:

- bound on the best value of the objective function

- প্রথমে breadth first manner এ explore করব। 3 ways:

  (i) queue

  (ii) stack

  (iii) least cost

- track the value of the best solution so far

3 condition for termination:

(i) not the best solution

(ii) not a feasible solution

(iii) no further option/choice is left.

Let, DP দিয়ে 0/1 Knapsack solve করেছি, branch and bound দিয়েও solve করেছি। Now what are the differences between them?

6/01/2025

Problems related to Branch & Bound

JOB ASSIGNMENT PROBLEM

$$
C = \begin{array}{c c}
\begin{array}{cccc} \text{job 1} & \text{job 2} & \text{job 3} & \text{job 4} \end{array} & \\
\begin{bmatrix} 9 & 2 & 7 & 8 \\ 6 & 4 & 3 & 7 \\ 5 & 8 & 1 & 8 \\ 7 & 6 & 9 & 4 \end{bmatrix} & \begin{array}{l} \text{person a} \\ \text{person b} \\ \text{person c} \\ \text{person d} \end{array}
\end{array}
$$

- Want to assign the jobs so that total cost is minimum.
- what is the lower bound?

Initially প্রতি person এর জন্য lowest cost টা কে lower bound ধরে নিব i.e, এরচেয়ে কম কখনো হবেনা।

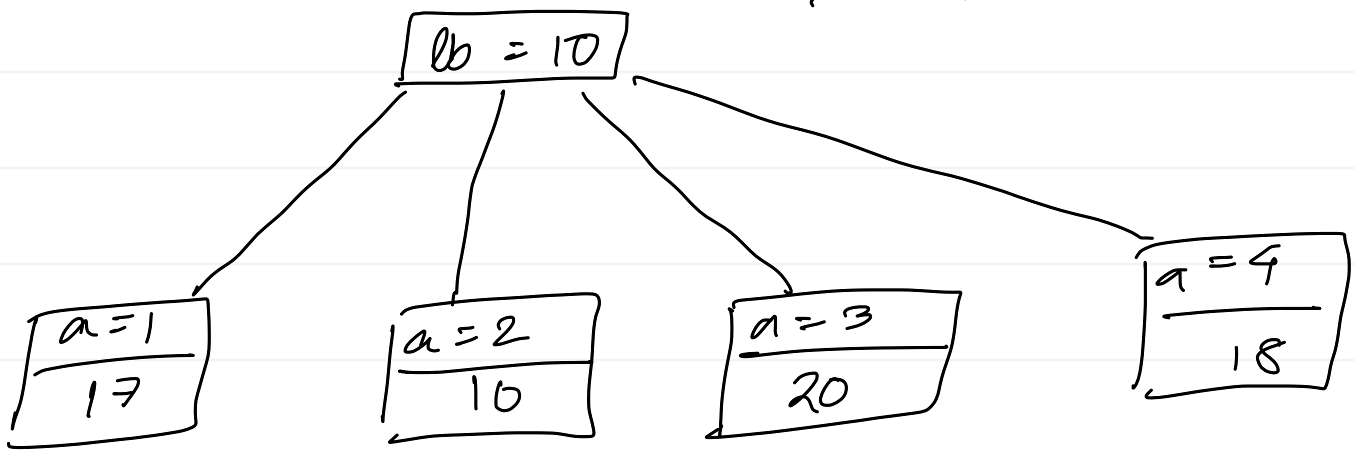
$\therefore$ $2 + 3 + 1 + 4 = 10$ (lower bound)

$\downarrow$

But এটা actually feasible না since same job cannot be assigned to two people (job 3 to person b and c)

i.e, lower bound নিই যে solution হবে
that is not guaranteed.

* lb = lower bound



* শুরুতে lb = 10 এর জন্য 2, 3, 1, 4. যখন a assigned job 1, তখন 9, 3, 1, 4. শুধু 1st টা update হবে, বাকিটা না, i.e এই column বাদে বাকিটা same.

when a → 2,

2, 3, 1, 4

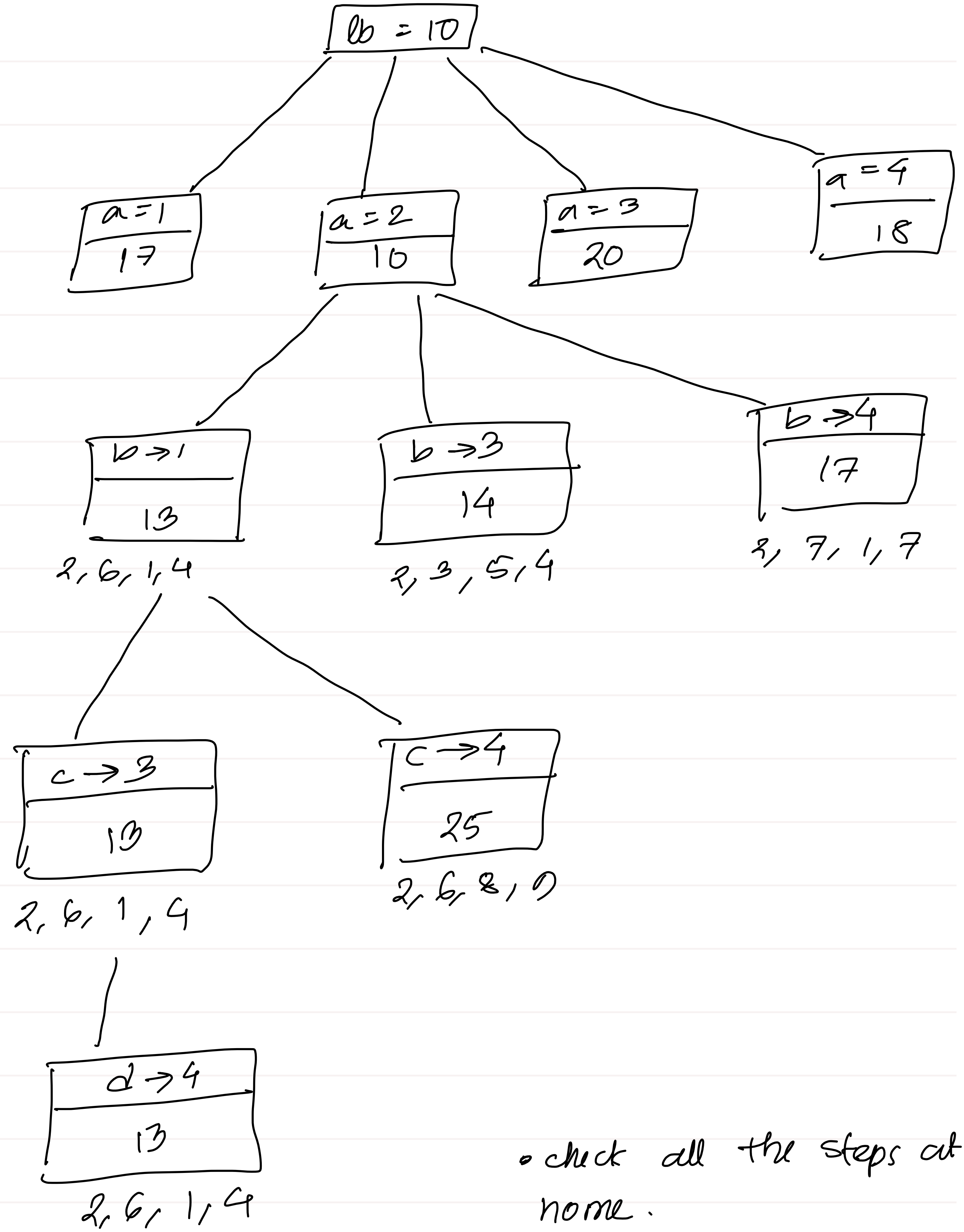
when a → 3,

7, 4, 5, 4

when a → 4

8, 3, 1, 6

Then যেটা closest to initial lower bound, সেখান থেকে next stage exploration করব।



• check all the steps at home.

Once we assign a job সেই column থেকে আর নিবনা। এই column বাদ দিয়ে each row থেকে minimum টা নিব। যেই job এখনো assign করিনি, সেই column এ conflict থাকতে পারে, that is not a problem.

KNAPSACK PROBLEM

fractional knapsack → greedy
0/1 knapsack → DP

Now, we will try to do 0/1 knapsack with branch and bound.

- It is an optimization problem.
  - maximization problem.

what is the initial upper bound?

item	weight	value	value/weight
1	4	\$40	10
2	7	\$42	6
3	5	\$25	5
4	3	\$12	4

The knapsack's capacity W is 10.

Initial upper bound:

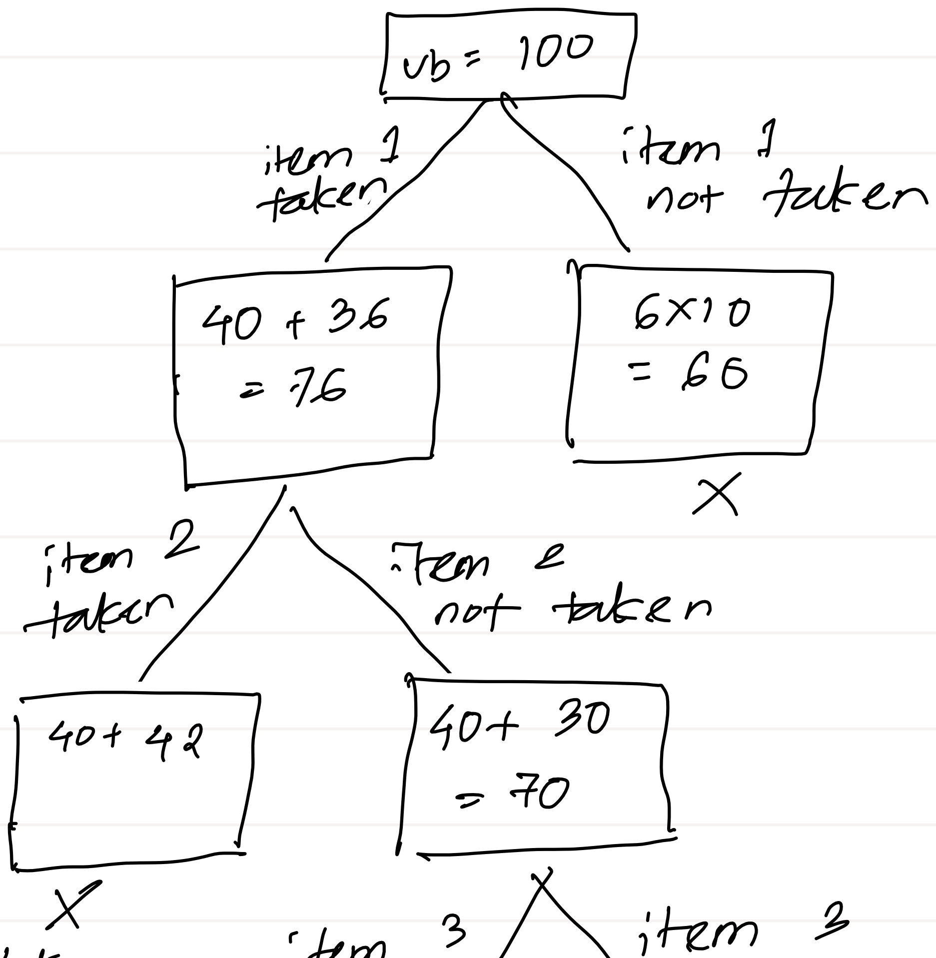
maximum per unit value × capacity = upper bound

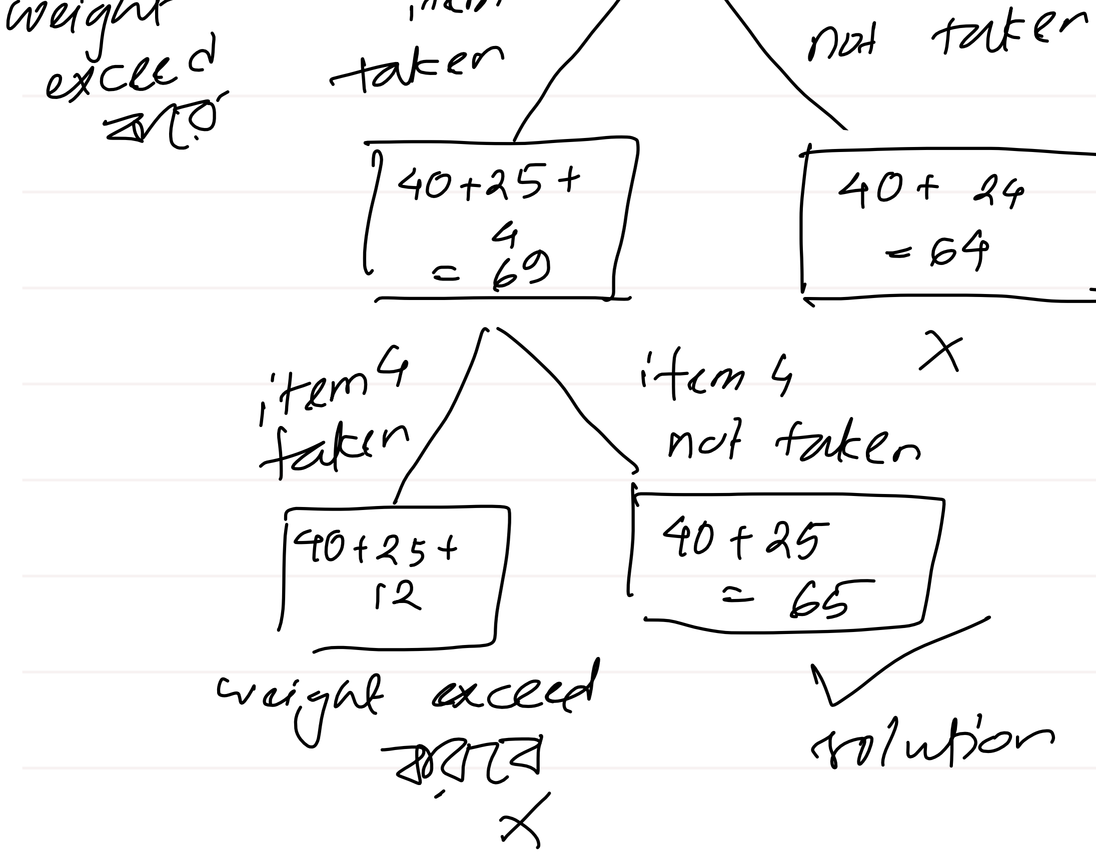
কাউকে choose করার পরে upper bound:

Already taken + Remaining capacity × max m per unit value remaining

$$ub = v + (W - w) \times \frac{v_i}{w_i}(\text{max.})$$

initially $v=0$ & $w=0$





* প্রতি node এ value, upper bound and weight রাখবে। এখানে just upper bound টা দেখানো।

ONLINE ALGORITHMS

- offline algorithm → we have all the data.
- online algorithm → we only have knowledge about the past, then we have to deal with the future.

- we measure the performance of an online algorithm using competitive ratio.

$$\text{competitive ratio}; \ CR(A) = \max_i \frac{A(i)}{OPT(i)}$$

→ ratio of what the online algorithm 'pays' to what the optimal offline algorithm 'pays'

SKI-RENTAL PROBLEM

- we don't know how many days we are going to ski for.

per day rent : 500 tk

one time buying cost : 1500 tk

optimal offline → we know কয়দিন যাবো, then we can take the decision accordingly. but here, we don't have that knowledge.

- online এ there could be a few different approaches:
  - (i) keep renting
  - (ii) buy শুরুতেই
  - (iii) কয়েকদিন rent, then buy on 3rd day

(i) ⇒

$$CR = \max\left(\frac{500}{500}, \frac{1000}{1000}, \frac{1500}{1500}, \frac{2000}{1500}, \frac{2500}{1500}, \dots\right)$$

$$= \infty$$

(ii) ⇒

$$CR = \max\left(\frac{1500}{500}, \frac{1500}{1000}, \frac{1500}{1500}, \frac{1500}{1500}, \dots\right)$$

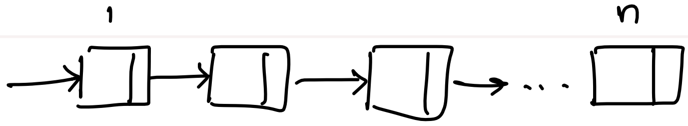
$$= 3$$

(iii) ⇒ $CR = \max\left(\frac{500}{500}, \frac{1000}{1000}, \frac{2500}{1500}, \frac{2500}{1500}, \dots\right)$

$$= 1.67$$

- Another problem: সব problem এর optimal solution আমার জানা নাই।

Linked List Search



কোনো element কে আমি search করে পাওয়ার পর we will shift it 2 places left i.e bring it closer so that if I request the same item repeatedly, I will get it faster each time.

Let, I am requesting nth item each time.

$$\therefore n + (n-2) + (n-4) + \dots + \underbrace{1}_{odd}/\underbrace{0}_{even} \approx \underbrace{a}_{\text{(something constant)}}n^2$$

General term: $(n-2k)$

Optimal solution:

একবার খুঁজে পাওয়ার পর তাকে একদম সামনে নিয়ে এসে রাখবে because we already know the entire search sequence.

$$\therefore n + 1 + 1 + \dots + 1 = 2n - 1$$

এখানে we can find a competitive ratio here.

$$= \frac{an^2}{2n-1} \approx \frac{n}{2}$$

However, let there is a random search sequence 1, 3, 2, 235, 5. এই ক্ষেত্রে we don't know the optimal solution. [Adversary situation]

However, there is a claim:

- For move to front strategy, CR is 2.

<u>Another claim:</u>

There is no strategy for which CR is less than 2.

Many variations:

(i) একটি element hit করার পর bring it to 1st position

(ii) last position

(iii) middle position

We are trying to find a worst case scenario. Adversary situation/ worst case এ next search element is always the last element of the linked list after each turn/ modification.

Adversary sequence, $\sigma = x_1, x_2, \ldots, x_n$

- আমার যেই sequence সেটার all possible permutations find out করতে হবে।

$O_1 \rightarrow$ cost of finding every element in permutation 1 (π_1)

$$\pi_1 \rightarrow \binom{x_1}{c_{11}} \binom{x_2}{c_{12}} (\quad) \ldots$$

$$\text{Average cost} = (O_1 + O_2 + \cdots + O_t)/t$$

[t is permutation]

- x_K কে খুঁজব in all possible permutations i.e, all possible positions হবে।

$\therefore$ cost of finding :

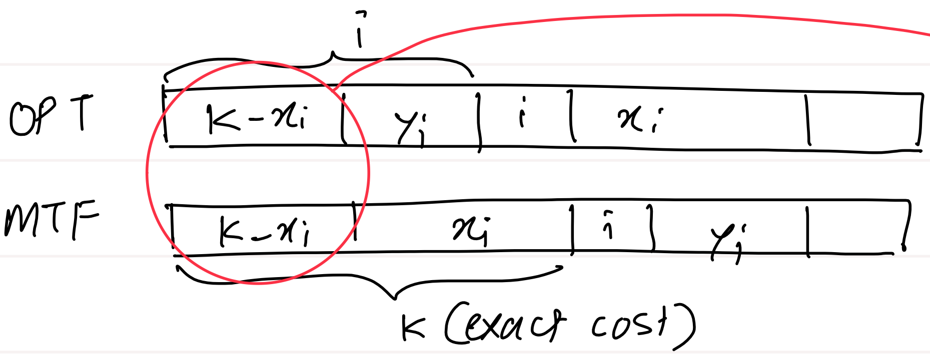
13.01.2025

Let us compare OPT (optimal) and MTF (move to front).

We assume that the optimal provides better results since it is optimal.

Sequence : 1, 2, 3, ... n

At step j, cost of MTF is k, OPT is i.



শুরুতে সমান সংখ্যক element তোমরা আগে access করে ফেলেছি। সেটার মাথাব্যথা নাই

Amortized cost : → chap 17 of CLRS

$$T(j) = k + INV(j+1) - INV(j)$$

exact cost + একটি operation চালানোর আগে পরে sequence এ যেই changes আসছে সেটার cost

$$= k + (k - x_i + y_i - \alpha) - (x_i + y_i)$$; $\alpha = i$ যতটুকু পরিমাণ সামনে

$$= 2k - 2x_i - \alpha \le 2k - 2x_i$$

$$\le 2i$$

(কিছু বাদ দিচ্ছি তাই $\le$)

∴ At most $2i$

Optimal এর i ই at most i. So, optimal এর দ্বিগুণ।

[প্রতি step এ at most double]

Then extend করে দেখাবো যে overall এই cost at most double.

<u>HW:</u>

Step $j+1$ এর execution এর পর sketch করব।

- Now, we will prove that there is no strategy whose CR is less than $2 - \frac{2}{n+1}$

Adversary situation: sequence এ always new list এর last item choose করতে বলে

Adversary cost $= n + n + \dots + n$

$= nm$ (If no. of requests is m and size of set is n)

Optimal algorithm একটা imagine করব → that knows the adversary sequence but we don't know the details of the algorithm.

We will make all possible permutations of my set. At least one will match the search sequence and it will give me the minimum cost. The minimum cost is obviously smaller than the average cost of all possible sequences.

∴ Upper bound of minimum cost → average cost

$\pi_1 = 1 \; 2 \; 3 \; \cdots \; n$

$\vdots$

π_t

In all permutations, average cost of finding 1st element be $C(i, 1)$

$$= \frac{(n-1)! \, (1+2+3+\cdots+n)}{n!}$$

$$= \frac{(n+1)}{2}$$

For finding m items,

$$\text{average cost} = \frac{m(n+1)}{2}$$

This is the upper bound of minimum cost.

$$\therefore \text{Competitive ratio} = \frac{n\cancel{m}}{\frac{\cancel{m}(n+1)}{2}}$$

$$= \frac{2n}{n+1}$$

$$= \frac{2n+2-2}{n+1}$$

$$= 2 - \frac{2}{n+1}$$

[Proved]

20/01/2025

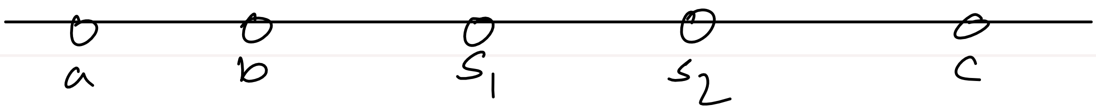
- Online algorithms are somewhat related to page replacement algorithm of OS.

- amortized analysis এর আরেকটা নাম হলো potential method.

22/01/2025

K- Server Problem

Let, there is a single road city. road এর দুইপাশে সব houses, offices etc. There is a company which provides internet service in this city. It deploys some vehicles/servers which solves connection problems.



a, b, c $\rightarrow$ clients

s_1, s_2 $\rightarrow$ server

- Cost is proportional to how much road a server traverses.

- We want to reduce the overall cost.

Greedy Approach

এখানে competitive ratio is infinite.

Let, request sequence :

cababab...

এখানে s_1, s_2 constantly move করবে and every step এ cost বাড়তেই থাকবে।

However, if we knew the entire sequence, একজনকে a তে বসায় রাখতাম, একজনকে b তে বসায় রাখতাম, এক্ষেত্রে cost আর বাড়বেনা।

copy this from
muhoun

Another approach:

Let, there are two service stations. Request যদি দুইটির ভিতরের থেকে আসে, তাহলে একভাবে treat করব, and যদি দুইটির বাইরে থেকে আসে, then আরেকভাবে treat করব।

এই class copy from meheoun

এই algorithm এ competitive ratio is constant.

Double Coverage:

- if the request falls outside the convex hull of the server, serve it with the nearest server. Otherwise, the request is in between two adjacent servers. In this case, move both these servers towards the request at equal speed until one server reaches it.

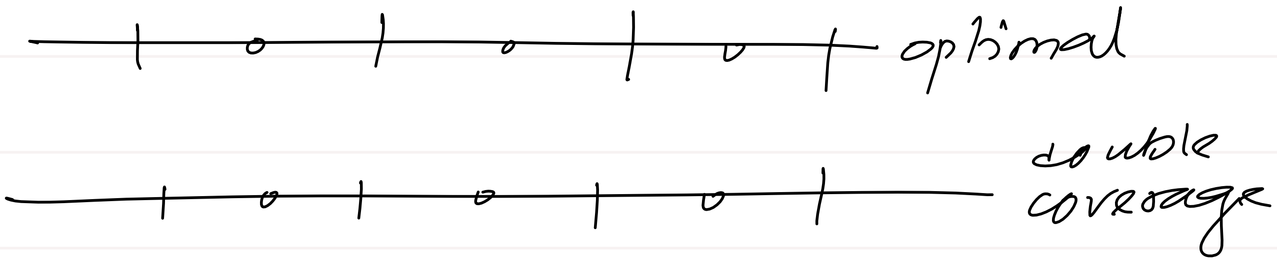
We will follow interleaving moving style.

Total cost of double coverage algorithm,

$$\sum_{DC} = \sum_{i \le j} d(s_i, s_j)$$

$\downarrow$

সবগুলো pair of station এর distance এর sum হচ্ছে double coverage.



দুইটা algorithm এর matching করলে minimum difference যেটা সেটা be Mmin.