Ahsan Habib Sir

Lecture-1

TOPIC NAME: Algorithm analysis & Design

DAY: Sunday

TIME:

DATE: 6 / 8 / 23

Algorithm:

Sequence of steps to perform a specific tasks.

Property:

① Input ② Output ③ Completeness ④ Finiteness

Complexity:

→ Nested loop = $O(n^2)$

→ constant time = $O(1)$

→ N element = $O(n)$

Graph Representation:

① Adj matrix

② Adj list

Graph Algorithm:

Graph $\xrightarrow{\text{Bfs / Dfs}}$ Spanning Tree

Edge Types:

1. Tree edge : New explore
2. Back edge :
3. Forward edge :
4. Cross edge :

Alwan Habib Sir

Lecture-2

TOPIC NAME : Complexity

DAY : Wednesday

TIME :

DATE : 9 / 8

Computer Program:

- Time → main concern
- Space
- Bandwidth

```
# int main () {
      int a=0;   → 1
      int b=0;   → 1
      int c= a+b;  → 2
```

$\underbrace{c=}_{1}\ \underbrace{a+b}_{1}$; $\underbrace{1+1+2}_{4 \text{ unit of time}}$

* Assign করলেই time লাগবে ।

```
for (int i=0; i<n; i++) { }   → 2n+2
     1, 4B    n+1   n
n=5;   Print(" ");
4B
}      a=10;          → n
       4B
                      ------
                       3n+2
```

$T(n) = 3n+2$

$S(n) = 12B$

TOPIC NAME : DAY : TIME : DATE : / /

□ int main() {

int n=5;

for(int i=0; i<n; i++) { } → 2n+2

$\underbrace{\text{int i=0}}_{1}$; $\underbrace{\text{i<n}}_{n+1}$; $\underbrace{\text{i++}}_{n}$

for(int j=0; j<n; j++) { } → n(2n+2)

$\underbrace{\text{int j=0}}_{1}$; $\underbrace{\text{j<n}}_{n+1}$; $\underbrace{\text{j++}}_{n}$

$\Rightarrow f(n) = 2n^2 + 2n + 2n + 2 = 2n^2 + 4n + 2$

n=10

□ for(int i=0; i<n; i+=2)

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

□ for(int i=0; i<n; i=i*i

int i=1; i<n; i=i*i

int i=2; i<n; i=i*i, i*2=i

$i = 2 = 2^1 = 2^{2^0}$

$i = 4 = 2^2 = 2^{2^1}$

$i = 16 = 2^4 = 2^{2^2}$

$i = 256 = 2^8 = 2^{2^3}$

$\Rightarrow \log_2(\log_2 n)$

$i = 2^0$

$i = 2^1$

$i = 4 = 2^2$

$i = 8 = 2^3$

$i = 16 = 2^4$

$\Rightarrow \log_2$

GOOD LUCK™

$i^3 = i * i * i$ } এও complex দরকার নাই।

2^3

2^9

2^{27}

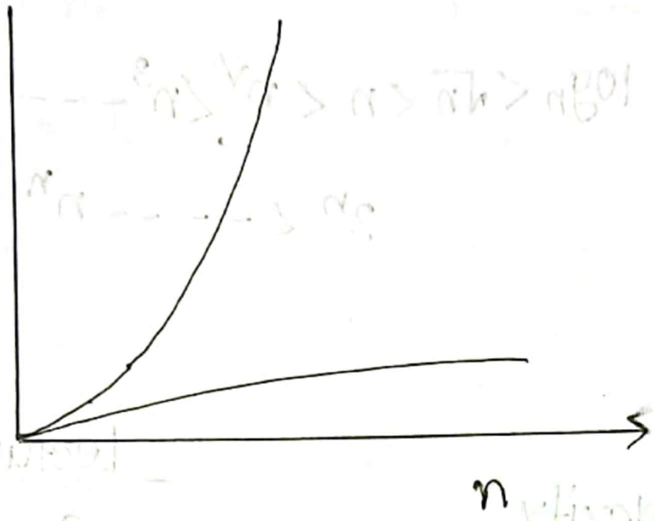
$\Rightarrow \log_3(\ )$

4) $4n+5 \rightarrow$ constant গুলা বাদ যাবে।

$n=1$	$4n$	5	$4n+5$
1	4	5	9
2	8	5	13
3	12	5	17
4	16	5	21
100	400	5	405

5) $n^2 + \underbrace{4n + 5}_{\times} \longrightarrow n^2$ consider করা লাগবে।

n^2	n	$4n$	5
1	1	4	5
4	2	8	



Cases:

1. Best $\Rightarrow O(1)$

2. Average $\Rightarrow O(\frac{n}{2})$

3. Worst $\rightarrow$ always worst case চিন্তা করি। $\Rightarrow O(n)$

Asymptotic Notation:

1. Best (Ω)
2. Worst (O)
3. Avg (θ)

$$c_1 \cdot g_1(n) < f(n) < c_2 \cdot g_2(n)$$
$\Downarrow$
Avg

Ex: $f(n) = 4n+5$ $g(n) = n$ $n = n_0$

$c \cdot g(n) > f(n)$

TOPIC NAME :

DAY :

TIME : DATE : / /

Complexity class:

$$1 < \text{------} < \log n < \sqrt{n} < n < n^2 < n^3 \text{----} < n^{100} < 2^n <$$
$$3^n < \text{-----} n^n$$

Lecture-3

Sunday

13/8/23

Ahsan Habib sir

Complexity

Book: Sartaj shani

Time complexity:

\# Algorithm 1.8 (Page-20)

Asymptotic Notation:

$$f(n) = O(g(n))$$

$$c \cdot g(n) \geq f(n)$$

\# sum ( ) { }

$$f(n) = 3n + 2$$

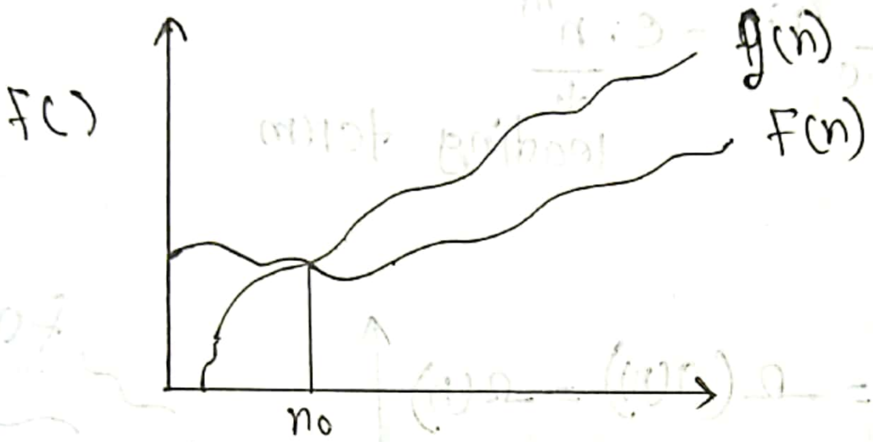
$$c \cdot n \geq 3n + 2 \Rightarrow 4n \geq 3n + 2$$

n → check করা লাগবে।

GOOD LUCK

$c = 4, n_0 = 1$

$f(n) = O(g(n)) = O(n)$



Example 1.11: $f(n) = 100n + 6 \rightarrow O(n)$ হবে কী না, হলে ও কার কর) ?

$100n + 6 \leq c \cdot n$

$\Rightarrow c = 101$

$100n + 6 \leq 101 \cdot n$

$n \geq 6,\ c = 101$ হবে

$\therefore f(n) = O(n)$

* g এর value minimum নিব।

→ leading term.

Theorem 1.2: $f(n) = a_m n^m + \cdots + a_1 n + a_0$

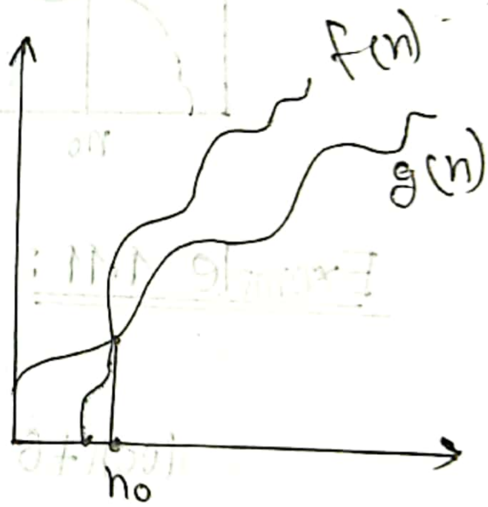
$$= \sum_{i=0}^{m} a_i n^i$$

$$= n^m \sum_{i=0}^{m} a_i n^{i-m}$$ → বড়টা consider করে।

$$= n^m \sum_{i=0}^{m} a_i = c \cdot \underbrace{n^m}_{\text{leading term}}$$

Omega Ω :

$$f(n) = \Omega(g(n)) = \Omega(n)$$



$c * g(n) \le f(n)$

$$3n \le 3n+2$$

$$c \cdot n^2 \le 3n+2$$

* ৩এর থেকে বড়টা নিবে।

Example-1.12 :

Theorem 1.3 :

Theta Θ :

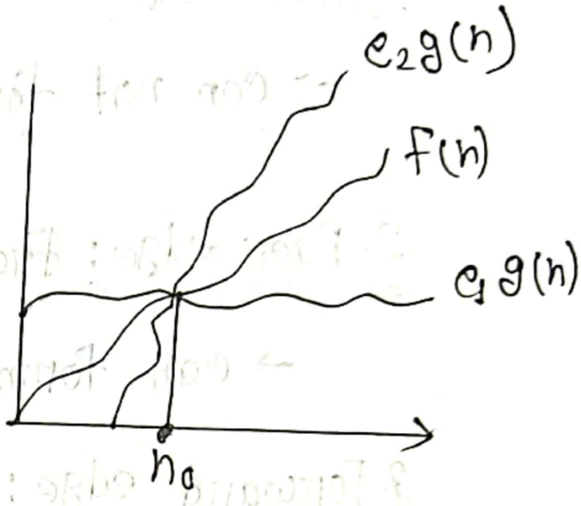
→ Precise value provide করে।

$$F(\Theta) = \Theta(g(n))$$

TOPIC NAME :   DAY :
TIME :   DATE : / /

⊛ $c_1 g(n) \le f(n) \le c_2 g(n)$ $f(n) = 3n+2$

$2n \le 3n+2 \le 4n$



Example 1.13 :

Theorem 1.4 :

Little o : $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

Example 1.14

Little Ω : $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$

Ahsan Habib Sir

Lecture-4

TOPIC NAME : DFS Edges

DAY : Sunday

TIME :

DATE : 20/8

DFS kind of edges:

1. **Tree edge:** encounter new vertex.
   → can not form cycle.

2. **Back edge:** From descendent to ancestor
   → can form cycle.

3. **Forward edge:** from ancestor to descendent.

4. **Cross edge:** between a tree or subtree.

BFS Traversal application:

1. Shortest path
2. Connected components.
   → কয়বার bfs call হচ্ছে, তাই।

→ Two color Problem.

DFS application:

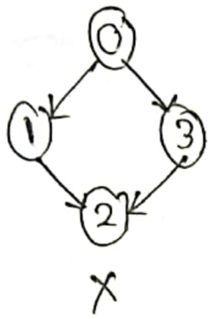
1. Finding cycle.
2. Topological sort
   → Node এর linear order.
3. SCC

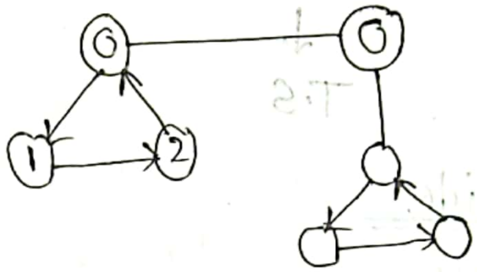
TOPIC NAME : DAY : TIME : DATE : / /

Topological sort : (Slide)

→ Indegree algorithm → Time descending algorithm

Strongly connected components : (Book- Cormen: বড় বই)





Ahsan Habib Sir

Critical Path

Lecture-5
Tuesday
22/8/23

Book : Algorithms unlocked

Critical Path in a PERT chart: Page-81

⇒ Maximum time = critical path

⇒ যার indeg 0 ও যার outdeg 0, তাদের মধ্যবর্তী summation of time = critical path

TOPIC NAME : DAY : TIME : DATE :

⇒ maximum path = minimum time

Shortest path in a directed acyclic graph :

DAG
↓
T.S

Scc → slide

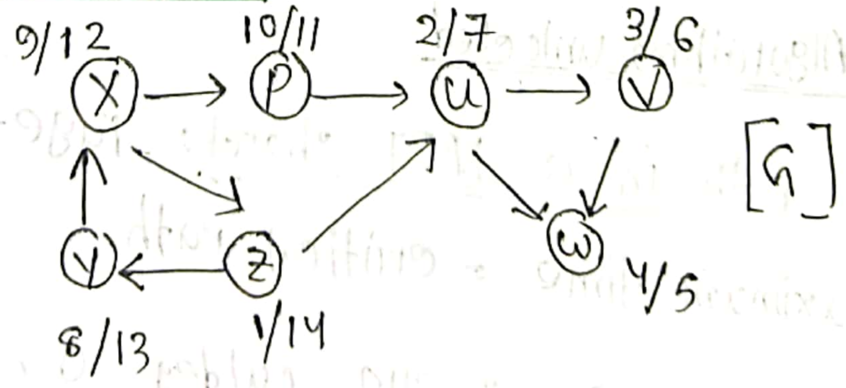
Ahsan Habib Sir

Network Flow

Lecture 6
Sunday
27/8/23

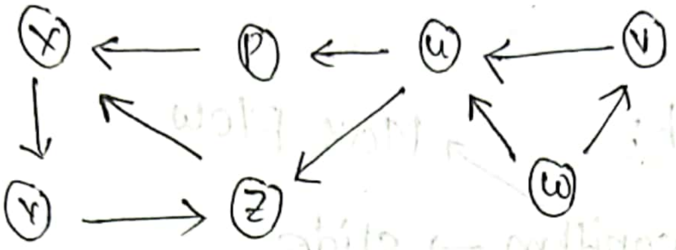
Book : Eva Trados → Algorithm Design

Scc :
↓
slide

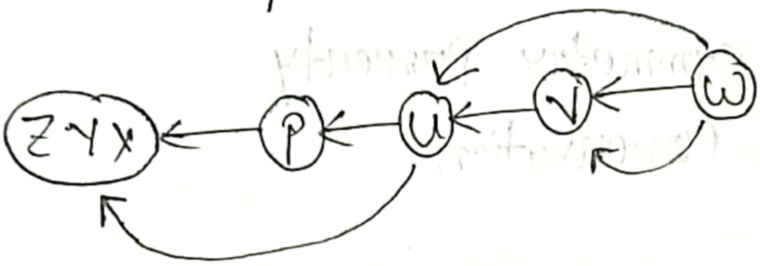


Z Y X P u V ω

G^T =

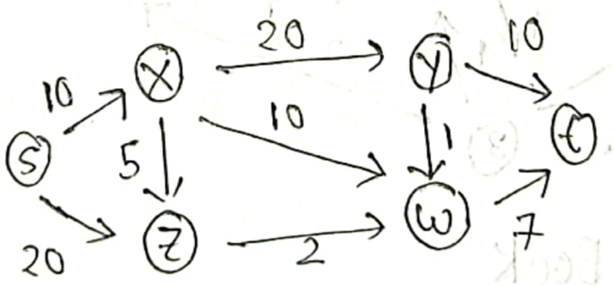


[G^T]



[Acyclic component]

Network Flow



* **Problem** : Maximum flow source to Destination.

Flow Network : Traffic flow + given capacity

Definition – Page 338 + 339

TOPIC NAME :

DAY :

TIME : DATE : /

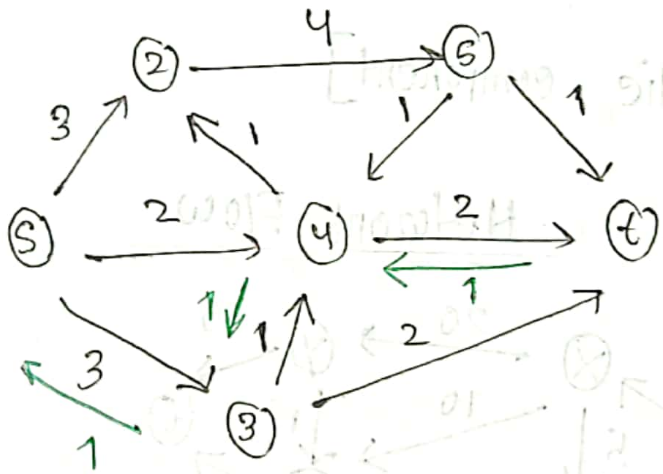
Defining flow:

The residual graph: → Max Flow

Ford Fulkerson Algorithm → slide

↳ skew symmetry property

↳ Flow conservation



Algorithm : Book

Lecture-7

TOPIC NAME: Network Flow

DAY: Tuesday

TIME:

DATE: 29/8/23

Book: Intro to Algorithm:

Ford Fulkerson:

Time complexity $O(f|E|)$

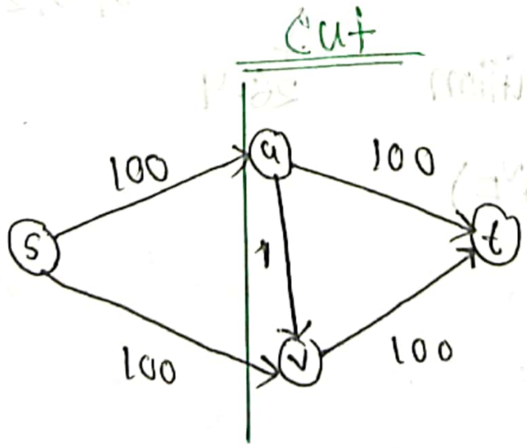
Problem: outer loop চলতেছে পর্যন্ত।

→ লম্বা path choose হলে, তাই loop বেশি হবে। worst case এইটা।

Edmond-Karp Algorithm:

→ Time complexity $= O(VE^2)$

→ BFS চলাতে, তাই longest path choose হবে না।



$L = \{s\}$ $R = \{u, v, t\}$

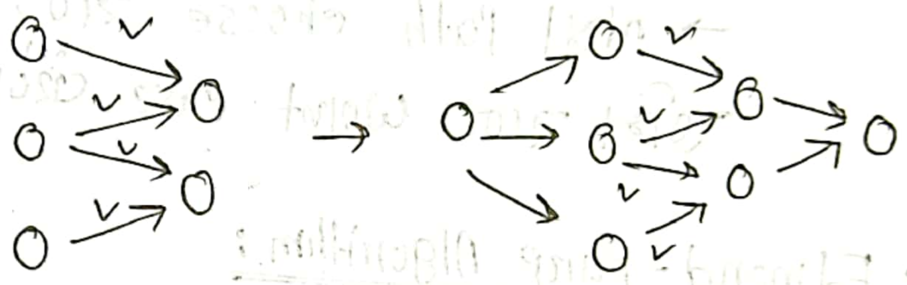
* Left to Right নেয়া লাগবে।
* cut এ s, t আলাদা পার্টে থাকবে।

TOPIC NAME : DAY : TIME : DATE :

*** minimum cut = max flow.

Maximum bipartite matching : Book + slide (pdf)

Push-relable algorithm : → Max flow algo



lecture-8
Sunday
3/9/23

Ahsan Habib Sir

Push Relable Algorithm

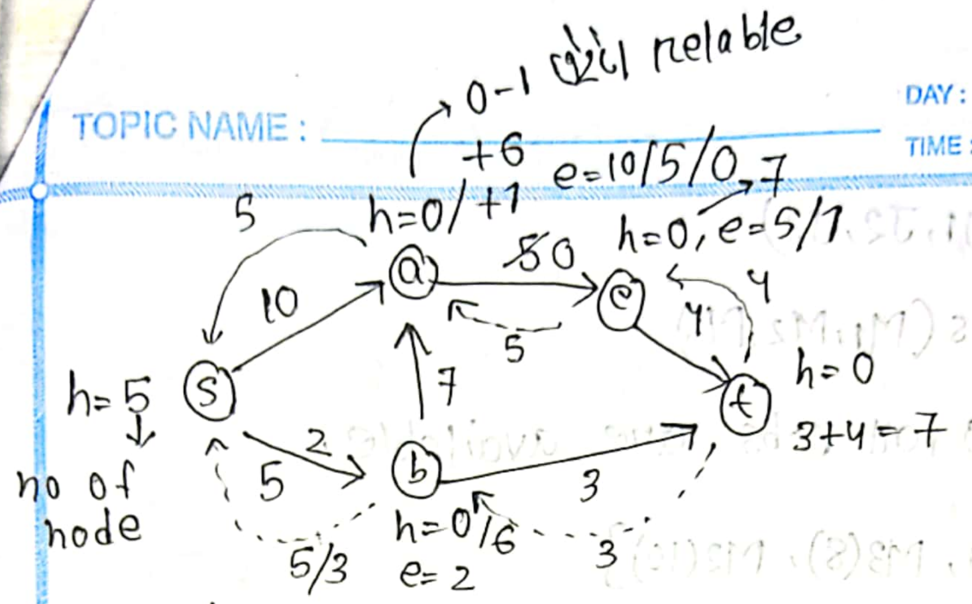
Book : Intro to Algorithm → 26.4

Complexity : $O(V^2E)$

2 operation :

(1) push (2) Relable

TOPIC NAME : DAY : TIME : DATE : / /



* যতক্ষণ না excess 0 হচ্ছে।

Height : h=5 ⟶ h=3

Excess : a ⟶ 10+7 − 5

* কোন node এর adjacent দিয়ে কাজ করার try করো।

26.5 :

Lecture-9
Sunday
10/9/23

Ahsan Habib Sir

Job shop Scheduling

Slide-6 :

⇒ Variant of scheduling algorithm.

Operation specific order → O_1, O_2 ---- O_n

⇒ Each operation for specific machine.

TOPIC NAME : DAY : TIME : DATE :

\# 3 Jobs (J1, J2, J3)

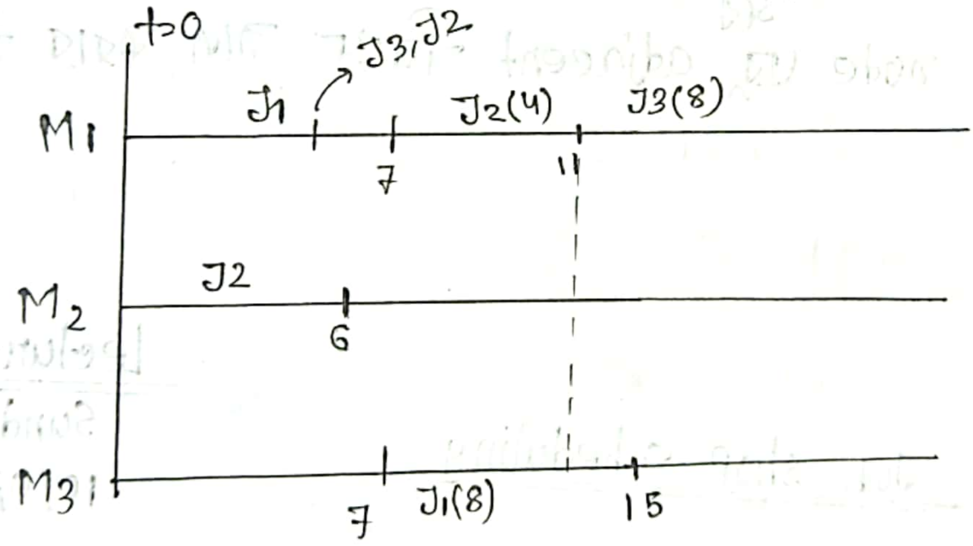
\# 3 machines (M1, M2, M3)

at time = 0, all Jobs are available.

$J1 = \{M1(7), M3(8), M2(10)\}$

$J2 = \{M2(6), M1(4), M3(12)\}$

$J3 = \{M1(8), M3(8), M2(7)\}$



Gantt chart

Lecture-10

TOPIC NAME: Bin Packing Algorithm

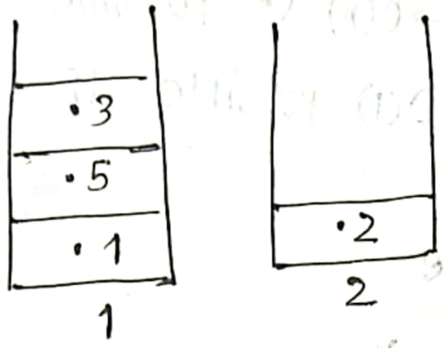
DAY: Sunday

TIME:

DATE: 17/9/23

⇒ Slide + PDF

.1, .5, .3, .2, .7, .9



⇒ First fit
⇒ Next fit
→ close to optimal soln

⇒ Best fit

⇒ Proof (Pdf → Page → 13)

↳ Page-14

Pdf → Page-15 ✗

↓ Slide ও shift

Bin Packing Heuristics :

Lecture-11

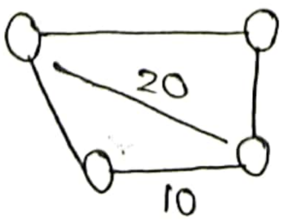
Tuesday

19/9/23

Ahsan Habib Sir

Approximation Algorithm

Traveling salesman Problem : → slide



TOPIC NAME : DAY : TIME : DATE :

Types complexity class -

* P class → Polynomial

2 5 0 1 → O(n) এ possible
→ O(1) possible না ।

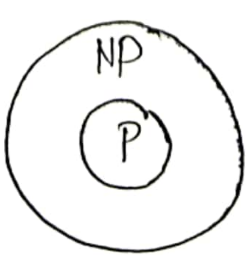
* deterministic machine
* Solvable and tractable

=> Ordered searching → O(log n)
=> Sorting – O(n log n)
=> String editing – O(mn)
=> Some exponential Algos.

* Nondeterministic → Assume করে নিব, then এর অনুযায়ী solve করা লাগবে।

NP class → Non deterministic P Time
→ Sol^n is hard to find.
→ Sol^n are easy to verify.

P-NP Relationship:



TOPIC NAME : DAY : TIME : DATE : / /

Satisfiability :

⇒ A model that makes the formula always true.

Satisfiability Problem (SAT)

CNF (SAT) → conjunctive normal form.

Rel^n bet^n SAT and Expon. Prb

NP Hard NP complete

↳ Book: Shani

Lecture-12
Sunday
1/10/23

Ahsan Habib Sir

NP-Hard, NP complete → que আসবে

⇒ chp-11
⇒ Book - Shani

\# P
NP
NP- hard
NP- complete.

Nondeterministic Algorithm :

A[n]

for (1-n)

⇒ O(n)

O(1) → exist করে না, যা nondeterministic machine এ process করা যায় না।

GOOD LUCK

TOPIC NAME : DAY : TIME : DATE : /

To write a nondeterministic algorithm:

1. choice (s) → arbitrary function
2. failure → kind of stopping condition
3. Success →

(failure and success: $O(1)$)

Example 11.1

$j \leftarrow$ choice $(1:n)$
if $A(j) = X$ then print (j): success
print (0); failure

Example 11.2: (***)

Example 11.3: Max clique problem

Example 11.4: Knapsack decision problem

Example 11.5:

Example 11.6:

যেকোনো হতে একটা আসবে, que must আসবে।

GOOD LUCK

Lecture- 13

TOPIC NAME : NP hard, NP complete

DAY : Sunday

TIME :

DATE : 15/10/23

Book : Shani

Example 11·7 : Knapsack Decision Problem

	x_1	x_2	x_3
$w=$	10	20	30
$p=$	100	200	300

$W=40$, $R=300$

for (1-n)

$x \leftarrow$ choice ( )

$S=\{x_1, x_3\}$

$w = 10+30$

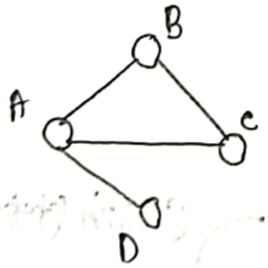
$P = 100+300$

Failure condition:

$\Rightarrow \sum w_i x_i > W$

$\Rightarrow \sum P_i x_i < R$

Clique Decision Problem (CDP)



for (1-K)

$t \leftarrow$ choice(1-n)

$S = S \cup t$

$S=\{A, D, C\}$

$\forall (i,j) \in S$

Failure Condition:

for (1-K)

$t \leftarrow$ choice(A/B/C/D)

$t \in S$

$S = S \cup t$

$S=\{A, A, C\}$

*** আমরা $P \subseteq NP$ লিখতেছি কিন্তু $P=NP$ লিখা possible কী না ?

NP (A_1 — A_{100})

$A_{10} \longrightarrow$ SAT

*** Cook's theorem

TOPIC NAME : DAY : TIME : DATE :

SAT: $\rightarrow$ exam এ আসবে

1/0

$F = x_1 \vee x_2 \wedge \overline{x_3}$

$\rightarrow$ Value এমন ভাবে assign করব x এর যাতে F always true হয়।

SAT 2 প্রকার:

(i) CNF $\longrightarrow \bigwedge_{c=1}^{k} c_i$ / $c_1 \wedge c_2$

$c_1 = x_1 \vee x_2 \vee x_4$

$c_2 = \overline{x_2} \vee x_3$

(ii) DNF $\longrightarrow \bigvee_{i=1}^{k} c_i$

$L_1 \propto L_2$

L_1 কে solve করতে পারলে L_2 কেও solve করা যাবে।

* CNF-SAT $\propto$ CDP $\rightarrow$ SAT

$F = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$ $\rightarrow$ graph (***)

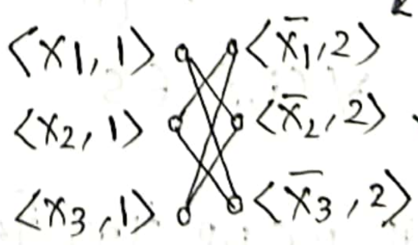
for (1-n)

$x_i \leftarrow$ choice (0/1)

$x_1 = 0$

$x_2 = 1$

$x_3 = 0$



$\langle x_2, 1\rangle$ — $\langle \overline{x_1}, 2\rangle$

$x_2 = T$ $x_1 = F$

$\rightarrow$ CDP

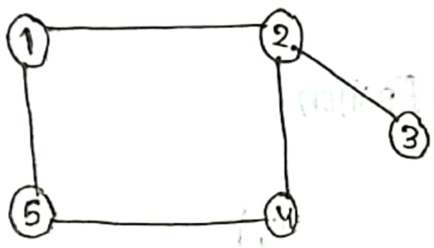
*** $x - \overline{x}$ edge হবে না

*** clique size 2, ৬টা clique আছে।

TOPIC NAME : DAY :
TIME : DATE : / /

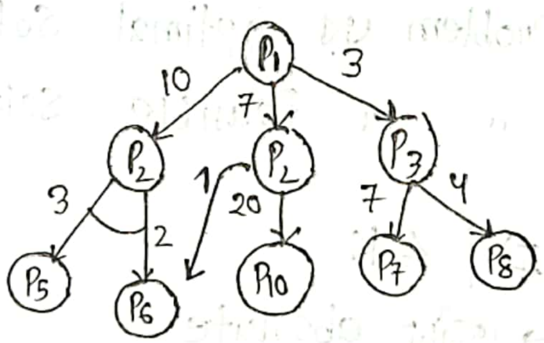
Node Cover Problem

→ Only Definition : Graph এ একটি Node এ যদি অপর Node গুলোর direct link থাকে।



AND/OR Graph

~~15~~
~~8~~
7



Ahsan Habib Sir

lecture-

TOPIC NAME: Approximation Algorithm   DAY: Sunday   TIME: DATE: 5/11

Book: shani (chp-12)

=> Feasible solution ⟶ close to optimal.

Three types-

① Absolute ② Function ③ Epsilon

P_1, P_2 $\qquad$ ω

ω_1, ω_2

=> $F^*(I)$ → এই Problem এর optimal solution.

$\hat{F}(I)$ → " " " Feasible solution.

=> $|F^*(I) - \hat{F}(I)| \leq K$

↳ একে absolute

Example 12.1:

$P_1, P_2, P_3 = 20, 10, 19$ $\quad n=3$

$\omega_1, \omega_2, \omega_3 = 65, 20, 35$ $\quad M=100$

$\quad\quad 1 \quad 0 \quad 1$

$65+20 \qquad 20+10$

$65+35 \qquad 20+19$

$$\overset{39}{\uparrow}$$

$|F^*(I) - \hat{F}(I)| \leq K$ ⟶ K যত ছোট হবে, ততো ভালো।

$|39 - 20|$

$19 \leq 10$

Example 12.2:

$\{P_1, P_2\} = \{2, r\}$ $\quad n=2$

$\{w_1, w_2\} = \{1, r\}$ $\quad M=r$

1, 0

If, $r = 1 \rightarrow 1, 0$

If, $r > 1 \rightarrow 0, 1$

Absolute Approximation:

Planar Graph coloring:

→ planar graph = 4 colorable

Lecture-15

Sunday

12/11/23

Ahsan Habib Sir

Hashing Technique

→ slide

A [2007001] = 15

121

A [2007121] = 20

A

B

email →

Roll (7) ⟶ F() ⟶ ind(3)

Al Ahsan Habib Sir

Lecture-16

TOPIC NAME : Summary

DAY : Tuesday

TIME :

DATE : 14 / 11 / 23

⇒ Prev year এর unknown ques দেখার দরকার নাই।

⇒ Hashing থেকে must ques.

#littleGiant_ 34