Name: Ayush Jain
SAP ID: 60004200132
Div: B
Computer Engineering

AOA

Assignment 1

Q.1) Explain Finding median using Divide and Conquer.

→ median of a list is its 50th percentile: half the number are bigger than it, half are smaller.

Method to find median:

1) Find the smallest element problem.
2) Pick any element v from a list S
3) Split a list S into 3 parts
   - elements small than V
   - those equal to V
   - elements greater than V
4) Narrow search into one of sub lists based on k.

For eg:

i) Let v=5 and S =

2	36	5	21	8	13	11	20	5	4	1

2) Split S in 3 parts

S_L =

2	4	1

S_V =

5	5

S_R:

36	21	8	13	11	20

$|S| = 11$, $|S_L| = 3$, $|S_V| = 2$

∴ Median will be located in S_R and search the lowest value of S_R

The algorithm is defined as:

$$\text{Selection}(S, k) = \begin{cases} \text{selection}(S_L, k) & \text{if } k \le |S_L| \\ v & \text{if } |S_L| \le |S_L| + |S_V| \\ \text{selection}(S_R, k - |S_L| - |S_V|) & \text{if } k > |S_L| + |S_V| \end{cases}$$

Time complexity:

Best case : T(n)+ T(n/2) + O(n)

Worst case: $n+(n-1)+(n-2)+\dots\frac{n}{2} = O(n^2)$

Average case:

we can split list into 3 quater size average

$T(n) \leq T(\frac{3n}{4}) + O(n)$

∴ Total execution time will be $T(n) = O(n)$

Q. 2 Maximize $z = 3x_1 + 2x_2 + 5x_3$

subject to $x_1 + 2x_2 + x_3 \leq 430$

$3x_1 + 2x_3 \leq 460$

$x_1 + 4x_2 \leq 420$

$x_1, x_2, x_3 \geq 0$

→ In standard form,

$z - 3x_1 - 2x_2 - 5x_3 + 0s + 0s_2 + 0s_3 = 0$

$3x_1 + 0x_2 + 2x_3 + 0s_1 + 0s_2 + 0s_3 = 460$

$x_1 + 2x_2 + x_3 + s_1 + 0s_2 + 0s_3 = 430$

$x_1 + 4x_2 + 0x_3 + 0s_1 + 0s_2 + s_3 = 420$

Sample Table:

Iteration number	Basic variables	Coefficients of x_1	x_2	x_3	s_1	s_2	s_3	RHS Solution	Ratio
0	z	-3	-2	-5	0	0	0	0	
x_3 enters	s_1	1	2	1	1	0	0	430	430/1 = 430
s_2 leaves	s_2	3	0	2	0	1	0	460	460/2 = 230
	s_3	1	4	0	0	0	1	420	420/0 = 0

Iteration number	Basic variables	Coefficient of x_1	x_2	x_3	s_1	s_2	s_3	RHS solution	Ratio
1	z	9/2	-2	0	[illegible]	3/2	0	1150	
x_2 enters	s_1	-1/2	2	0	1	-1/2	0	200	200/2 = 100
s_1 leaves	x_3	3/2	0	1	[illegible]	1/2	0	230	230/0 = 0
	s_3	1	4	0	0	0	1	420	420/4 = 105

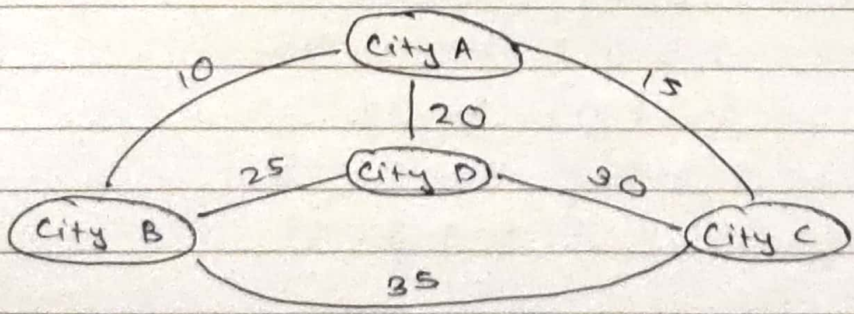
Basic variable	Iteration number	Coefficients of x_1	x_2	x_3	s_1	s_2	s_3	RHS solution	Ratio
2	z_1	4	0	0	1	2	0	1350	
	x_2	-1/4	1	0	1/2	-1/4	0	100	
	x_3	3/2	0	1	0	1/2	0	230	
	s_3	2	0	0	-2	1	1	-210	

All z are positive.

$\therefore$ The solution is $x_1 = 0$, $x_2 = 100$, $x_3 = 230$

$\therefore Z_{max} = 1350$

Q. 3) Solve the Travelling salesman Problem for the graph.



Scanned with CamScanner

→ Above graph can be represented by:

	A	B	C	D
A	0	10	15	20
B	10	0	35	25
C	15	35	0	30
D	20	25	30	0

For $i \in S$,

$$g(i,S) = \min_{j \in S} \{ C_{ij} + g(j, S - \{j\}))$$

$g(i, \phi)$ = means start at node i and end at 1 with no vertices in between.

Iteration 1:

$g(B, \phi) = C_{BA} = 10$

$g(C, \phi) = C_{CA} = 15$

$g(D, \phi) = C_{DA} = 20$

Iteration 2:

$g(B, \{C\}) = C_{BC} + g(C, \phi) = 35 + 15 = 50$

$g(B, \{D\}) = C_{BD} + g(D, \phi) = 25 + 20 = 45$

$g(C, \{B\}) = C_{CB} + g(B, \phi) = 35 + 10 = 45$

$g(C, \{D\}) = C_{CD} + g(D, \{\phi\}) = 30 + 20 = 50$

$g(D, \{B\}) = C_{DB} + g(B, \phi) = 25 + 10 = 35$

$g(D, \{C\}) = C_{DC} + g(C, \phi) = 30 + 15 = 45$

Iteration 3:

$$g(B, \{C, D\}) = \min \left\{ \begin{array}{l} C_{BC} + g(C, \{D\}) \Rightarrow 35 + 50 = 85 \\ C_{BD} + g(D, \{C\}) \Rightarrow 25 + 45 = 70 \end{array} \right\} = 70$$

$$g(C,\{B,D\}) = \min \begin{cases} C_{CB} + g(B,\{D\}) \Rightarrow 35+45 = 80 \\ C_{CD} + g(D,\{B\}) \Rightarrow 30+35 = 65 \end{cases} = 65$$

$$g(D,\{B,C\}) = \min \begin{cases} C_{DB} + g(B,\{C\}) \Rightarrow 35+50 = 85 \\ C_{DC} + g(C,\{B\}) \Rightarrow 30+45 = 75 \end{cases} = 75$$

Iteration 4:

$$g(A,\{B,C,D\}) = \min \begin{cases} C_{AB} + g(B,\{C,D\}) = 10+70 = 80 \\ C_{AC} + g(C,\{B,D\}) = 15+65 = 80 \\ C_{AD} + g(D,\{B,C\}) = 20+75 = 95 \end{cases} = 80$$

Shortest path : $A \rightarrow B \rightarrow D \rightarrow C \rightarrow A$

OR

$A \rightarrow C \rightarrow D \rightarrow B \rightarrow A$

Q.4) Write short note on Asymptoic Notations:

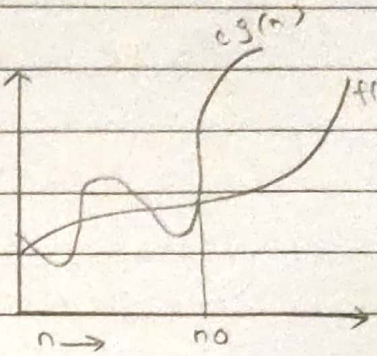
→ Asymptoic notations are a mathematical tool to find time or space complexity of an algorithm without implementing it in a programming language. These measure is independent of machine-specific constants.

There are mainly 3 asymptotic notations:

1) Big O - notation
2) Omega notation
3) Theta notation

(i) Big O Notation:

Big O Notation defines upper bound for the algorithm. It means the running time of algorithm cannot be more than its asymptotic upper bound. It gives worst-case complexity of an algorithm. let $f(n)$ and $g(n)$ are two non-negative functions indicating running time of 2 algorithms. we say $g(n)$ is upper bound of $f(n)$ if there exist some positive constants c and n_0 such that $0 \leq f(n) \leq c \cdot g(n)$ for all $n \geq 0$. It is denoted as $f(n) = O(g(n))$.



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

For eg: $T(n) = 3n+2$

$0 \leq f(n) \cdot c g(n)$

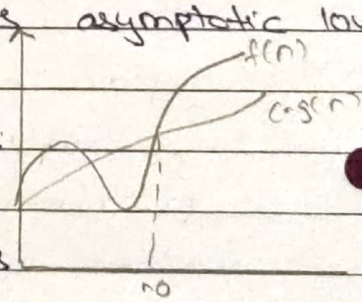
$0 \leq 3n+2 \leq c \cdot g(n)$

$0 \leq 3n+2 \leq 3n+2n$

$\therefore 0 \leq 3n+2 \leq 5n$

(ii) Big Omega Notation :

Big omega notation defines lower bound of algorithm. Running time of algorithm cannot be less than its asymptotic lower bound. It provides best case complexity. let $f(n)$ and $g(n)$ be 2 non-negative functions. The function $g(n)$ is lower bound of function $f(n)$ if there exist some positive constants c and n_0 such that $0 \leq c \cdot g(n) \leq f(n)$ for all $n \geq n_0$. $f(n) = \Omega(g(n))$.



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

For eg:

$T(n) = 3n+2$

$0 \leq c \cdot g(n) \leq f(n)$

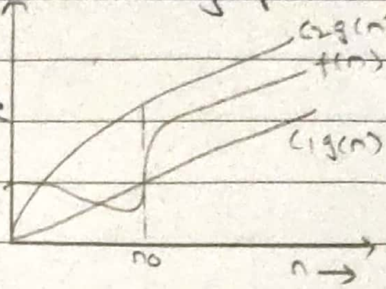
$0 \leq c \cdot g(n) \leq 3n+2$

$0 \leq 3n \leq 3n+2$

(iii) Big Theta Notation :

Theta notation defines tight bound for algorithm. Running time should be less than or greater than its asymptotic tight bound. It provides average case complexity. let $f(n)$ and $g(n)$ be 2 non-negative functions. The function $g(n)$ is tight bound of function $f(n)$ if there exist some positive constants c_1, c_2 and n_0 such that $0 \le c_1 g(n) \le f(n)$; $f(n) \le c_2 g(n)$ for all $n \ge n_0$.

$f(n) = \Theta(g(n))$.



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

For eg:

$T(n) = 3n+2$

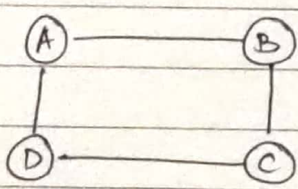
$0 \le c \cdot g(n) \le f(n) \le c_2 g(n)$

$0 \le c_1 g(n) \le 3n+2 \le c_2 g(n)$

$0 \le 3n \le 3n+2 \le 5n$ for all $n \ge 1$

Q.5) Explain Graph colouring Algorithm with help of an example?

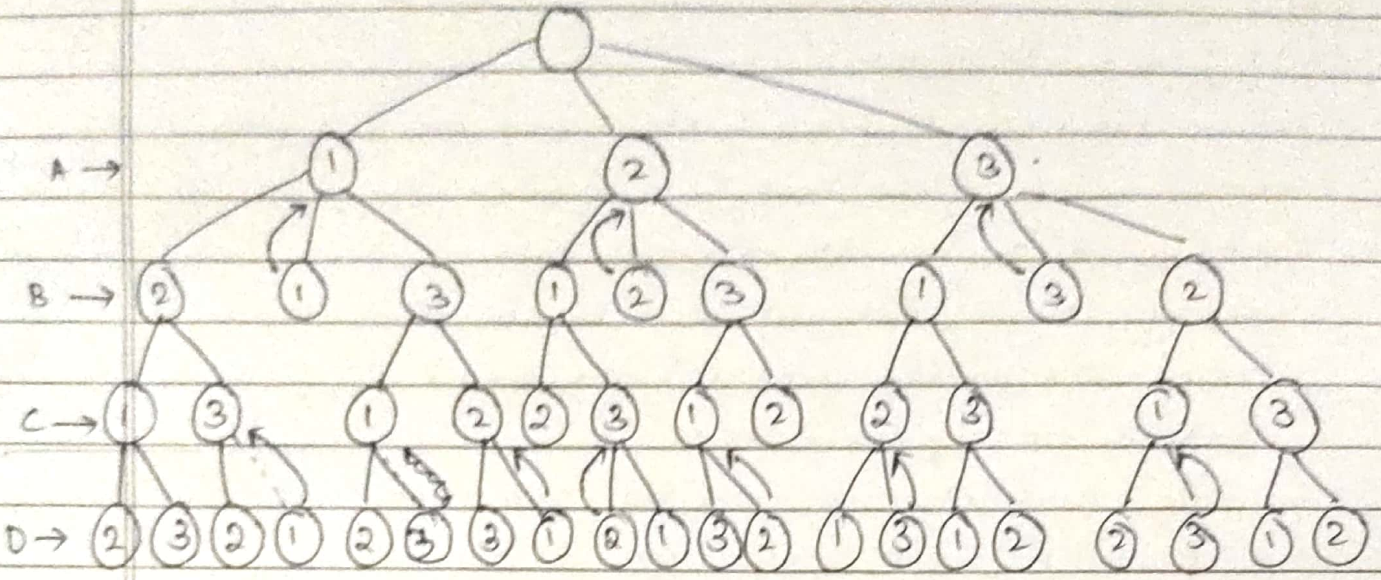
→



let there be 3 colours

$m = 3$

State space tree : (specifies all possible combination)



If we assign colour 1 to vertex A then colour 1 cannot be assigned to B and D and thus we back track and assign next colour. In next step B is assigned some different colour 2 or 3. Now vertex A is already coloured and C is neighbourhood of B, so C cannot be assigned colour 3 if B is assigned colour 2 and vice-versa. The process goes on.

In this way using back tracking method graph colouring algorithm is solved.