CS - 771

Artificial Intelligence

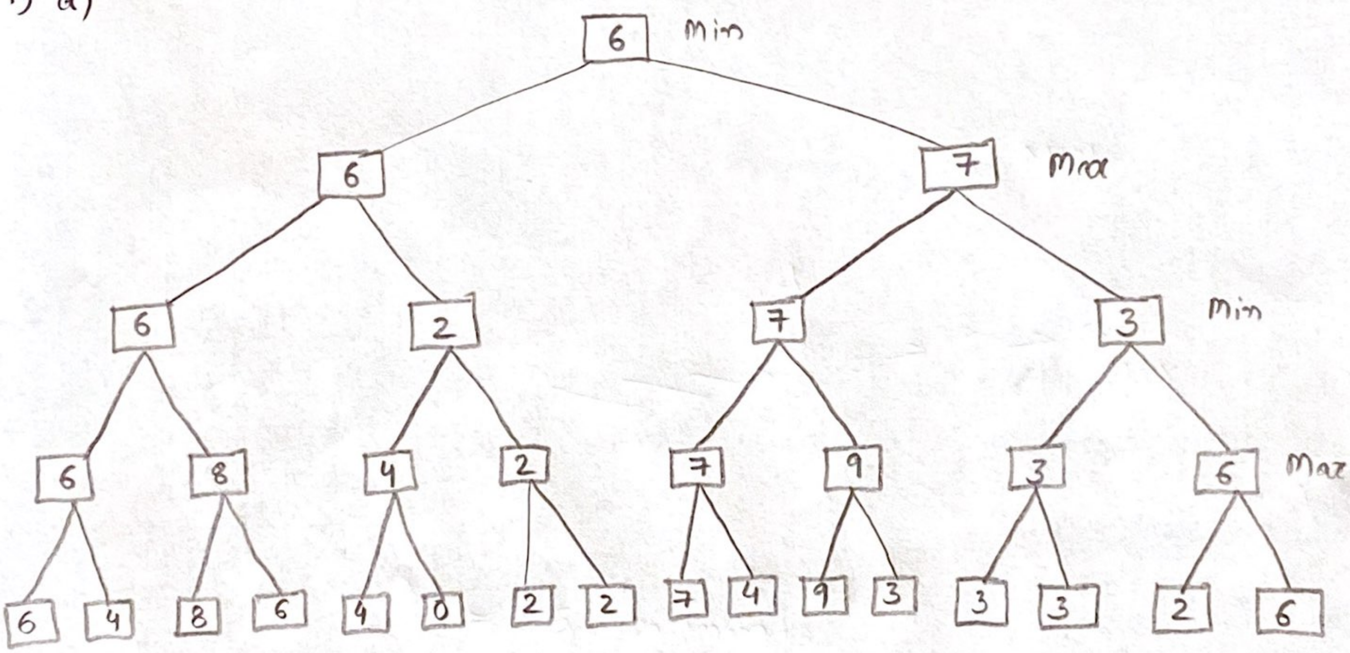
30th March, 2020

Spring 2020, Mid term

Chakradhar Reddy Demuri

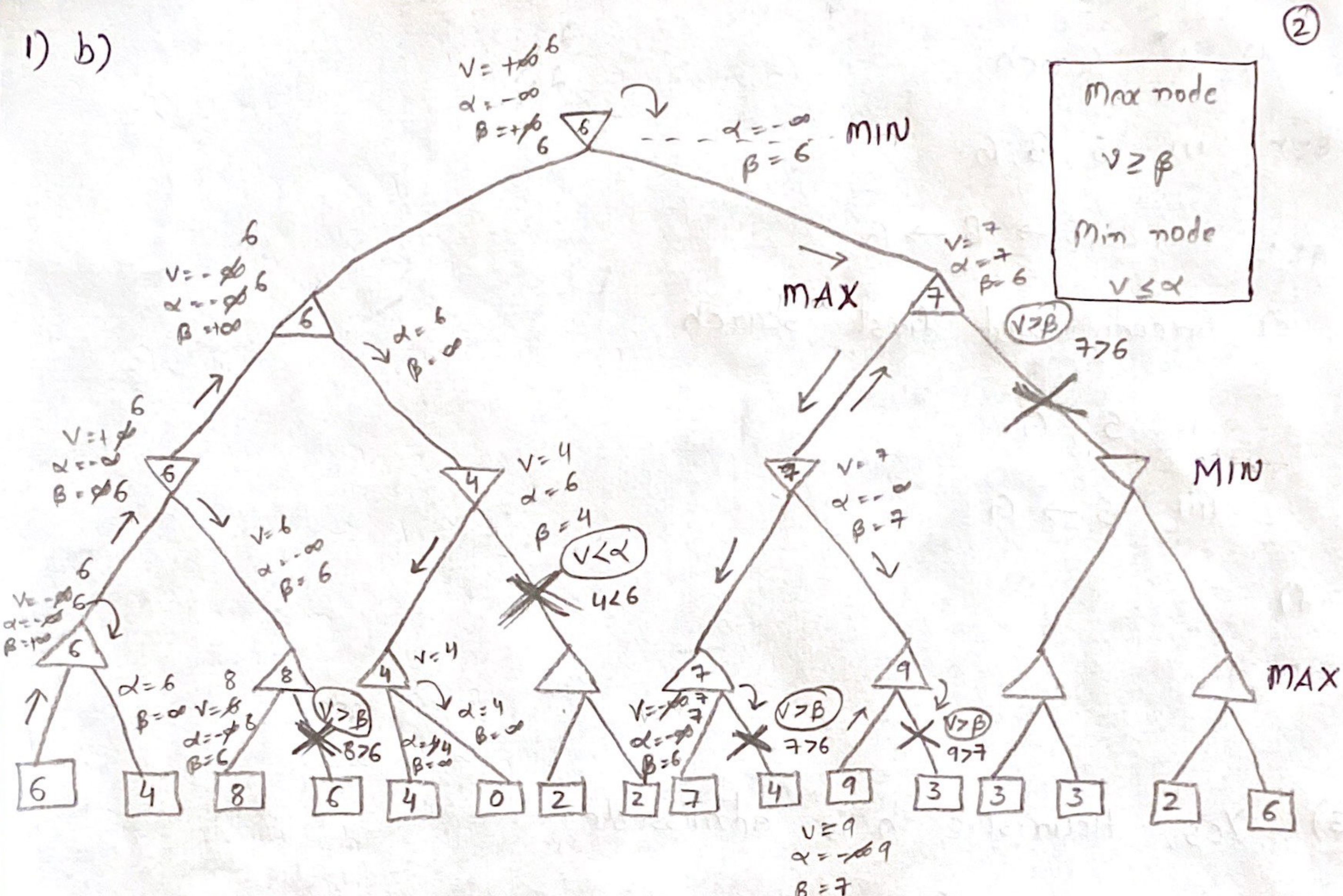
E949F496

1) a)



The value at the root is 6, using minimax algorithm.

1) b)



Alpha - Beta pruning = 5 prunings

2) a) BFS (Breadth first Search)

(i) S, A, B, C, G

(ii) S → G

b) DFS (Depth first search)

(i) S, A, G

(ii) S → A → G

~~c) Uniform cost search~~

(i) S, ~~C, B, G,~~ A

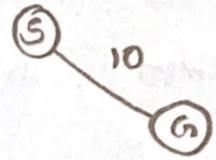
(ii) ~~S → B → G~~ ~~S → C → G~~

~~d) A* search~~

~~(i) S, B, G~~

~~(ii) S → B → G~~

e) Greedy best first search



(i) S, G

(ii) S → G

f) Yes, A* search return the optimal path.

Among all optimal algorithms that start from the same start node and use the same heuristic h, A* expands the minimal no. of paths.

- It is optimal when branching factor is finite and are costs are strictly positive

3) Yes, heuristic 'h' is admissible

$$h_K(n) \leq h^*(n)$$

$$h_K(n) \leq \frac{3}{4}\left(\alpha_1 h^*(n) + \alpha_2 h^*(n) + \alpha_3 h^*(n) + \cdots\right) + \frac{1}{4} \max\{h_1(n), h_2(n), \ldots h_K(n)\}$$

$$h_K(n) \leq \frac{3}{4} h^*(n)\{\alpha_1 + \alpha_2 + \alpha_3 + \cdots + \alpha_n\} + \frac{1}{4} h^*(n)$$

$$\because h^*(n) = \max\{h_1(n), h_2(n), \ldots h_K(n)\}$$

$$h_K(n) = \frac{3}{4} h^*(n) + \frac{1}{4} h^*(n) = h^*(n)$$

$$\because \alpha_1 + \alpha_2 + \dots + \alpha_n = 1$$

$$h_K(n) = h^*(n)$$

Hence, 'h' is admissible.

6) Goal test should be performed when node is popped from the queue if, you care about finding the optimal path and your search space may have both short expensive and long cheap paths to a goal.

- Guard against a short expensive goal
- Eg:- Uniform cost search with variable step costs.

4) $f(n) = a \times depth(n) - b \times depth(n)$

a, b are positive constants

$= (a-b)\, depth(n)$

a) $a > b$

consider $a = 4,\ b = 3$ i.e $(a > b)$
$4 > 3$

$n = 5$

$f(n) = (a-b)\, depth(n) = (4-3)\, depth(5)$

$f(5) = depth(5)$

It corresponds to Breadth first search (BFS)

b) $b > a$

consider $a = 3,\ b = 4$ (i.e $b > a$)
$4 > 3$

$n = 5$

$f(n) = (a-b)\, depth(n) = (3-4)\, depth(5)$

$f(5) = (-1)\, depth(5)$ ($\because$ negative)

$\therefore$ It corresponds to Depth First search (DFS)

5) Simulated Annealing

n_2 node has the higher probability of being selected to be the next current node

Let us consider for instance $n_1 = 0.3$, $n_2 = 0.4$

current value $= 0.7$ and $T = 0.8$

$\Rightarrow$ a) n_1.value < current.value $\Rightarrow 0.3 < 0.7$

b) n_2 value < current.value $\Rightarrow 0.4 < 0.7$

c) $|n_1.\text{value} - \text{current.value}| > |n_2.\text{value} - \text{current.value}|$

$$|0.3 - 0.7| > |0.4 - 0.7|$$

$$|-0.4| > |-0.3|$$

$$0.4 > 0.3$$

$\therefore$ from the algorithm $\Delta E \leftarrow$ next.value − current.value

$\therefore$ for $n_1 \Rightarrow e^{\Delta E/T} = e^{-0.4/0.8} = e^{-1/2} = 1/e^{1/2}$

$= 1/e^{1/2} = 0.6$

for $n_2 \Rightarrow e^{\Delta E/T} = e^{-0.3/0.8} = e^{-3/8} = 1/e^{3/8}$

$= 1/1.455 = 0.68$

$\therefore$ n_2 has higher probability so,

n_2 will be next node.

7)

a) True

b) False

c) ~~True~~ False

d) False

e) True

f) True

g) ~~False~~ True

h) True

i) True

j) False

k) True

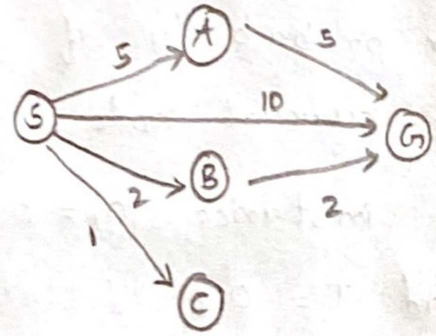
l) True

m) False

n) False

o) True

2) c) Uniform Cost Search



Frontier - F

Visited - V

V: S

F:

A	B	C
5	2	1

V: S → C

F:

A	B
5	2

V: S → C → B

F:

A	G
5	4

V: S → C → B → G

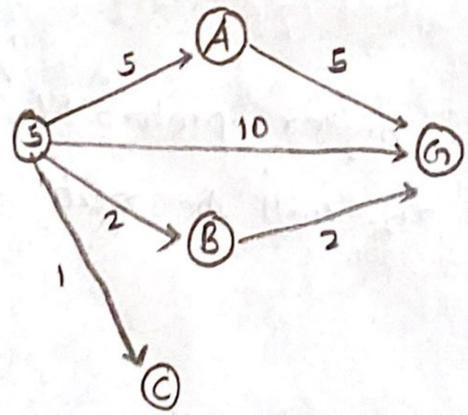
F:

G
4

(i) S, C, B, G

(ii) S → B → G

d) A* Search



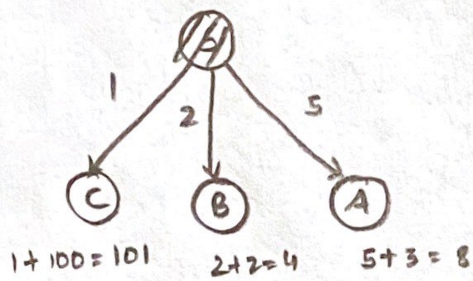
Node	h
S	4
A	3
C	100
G	0

Frontier : ○ — F

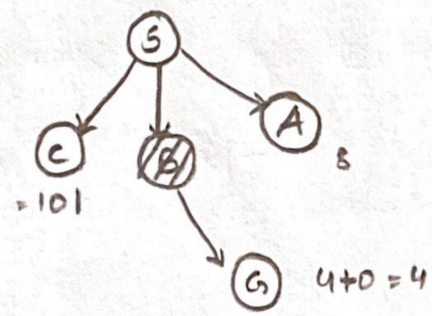
Visited : — V

step - 1

step - 2



step - 3



(i) S → B → G

(ii) S → B → G