Question 1

DATE

a) a) Fully observable since we can see the entire map.
b) It is deterministic since here the agent's current state and the action selected can determine the next state / change in environment
c) Sequential because the action taken will impact the next state.
d) Static because the map will not change.
e) Discrete because number of state in the environment is finite.
f) Single agent because there is only one agent.

b) 1). COVID-19 positive patient tracking system agent :-

→ Performance :- how fast it can track the patients, optimality of the search / tracking.
→ Environment :- 10/20 feet area surrounding the agent.
→ Actuators :- notification to the agent / alarm.
→ Sensors :- GPS for tracking the patients defi devices, other sensors.

II) Exploring the surface area of MARs.

→ Performance :- movability on rough surface, camera quality.
→ Environment :- the surface of Mars / Environment of MARs.
→ Actuators :- accelerator, brakes, etc.
→ Sensors :- IR sensor, video, sonar sensor, engine sensors and GPS.

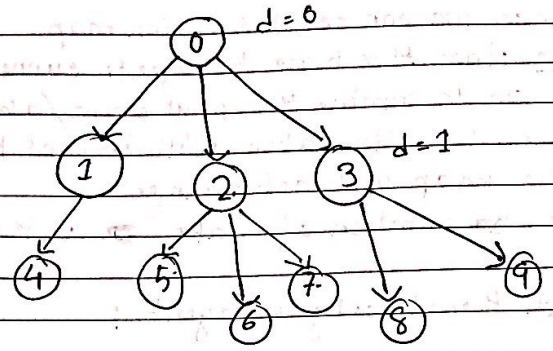
Typo

DATE

Question 2

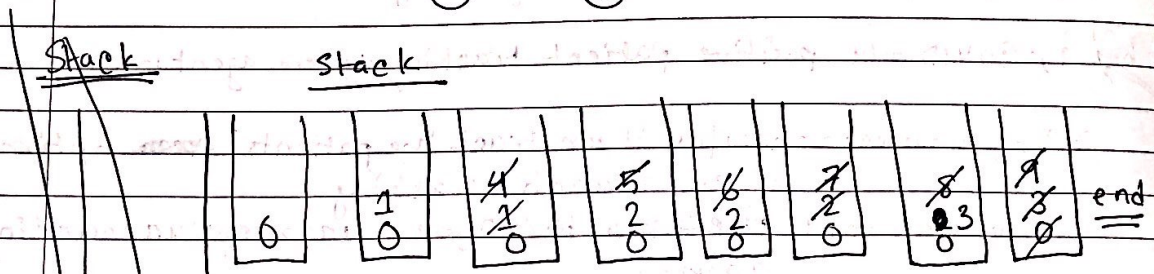
depth limit = 2.

i)



Stack

stack



Here $d <$ depth of goal node so this is incomplete.

$b = 5, m = 10$

ii) Total number of nodes generated = $\sum (b^m)$

$\Rightarrow b^1 + b^2 + b^3 + b^4 + b^5 + b^6 + b^7 + b^8 + b^9 + b^{10} =$

$= 12207030.$

Total time taken:-

1000000 node → generated in 1 sec.

1 node → " " $\frac{1}{1000000}$

12207030 → " " $\frac{12207030}{1000000} = 12.20703$ secs.

DATE

total space required :-

$O(bm + 1)$

$\Rightarrow (5 \times 10) + 1 \times 1000$ bytes

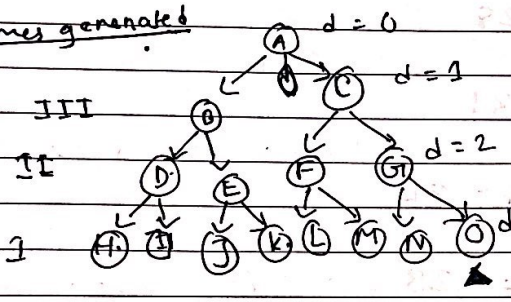
$= 5100$ bytes

1000 byte $\rightarrow$ 1 kb.

5100 " $\rightarrow \frac{5100}{1000} = 5.1$ kilobytes.

(iii) For the worst-case :-

no. of times generated



Worst case → if O is the goal node. because then it has to traverse all the nodes to reach O.

Let b be the branching factor, & d is the depth.

So,

$N(IDS) = db + (d-1)b^2 + \cdots + (1)b^d$.

$\therefore O(b^d)$ ← is the time complexity.

~~db → d=1 is generated~~

db → at depth = 1, it is generated d number of times.

$(d-1)b$ → at depth = 2, it is generated $(d-1)$ number of times.

$(1)b^d$ → at depth = d, it is generated 1 time only.

Typo

Question 3

DATE

a) All the path cost from A to H

A → B → E → H = 23

A → B → F → H = 25

A → C → F → H = 22

A → C → G → H = 20

A → D → G → H = 28

A → D → G → F → H = 31

A → D → G → C → F → H = 36

A → C → G → F → H = 23

A → C → F → B → E → H = 30

A → B → F → G → H = 26

A → B → F → C → G → H = 31

A → C → F → G → H = 23

~~A → D → G → C → F → B → E → H = 44.~~

A → D → G → C → F → B → E → H = 44.

A → D → G → F → B → E → H = 39

∴ The smallest path cost from A → H is 20

Heuristic of A is, h(A) = 12

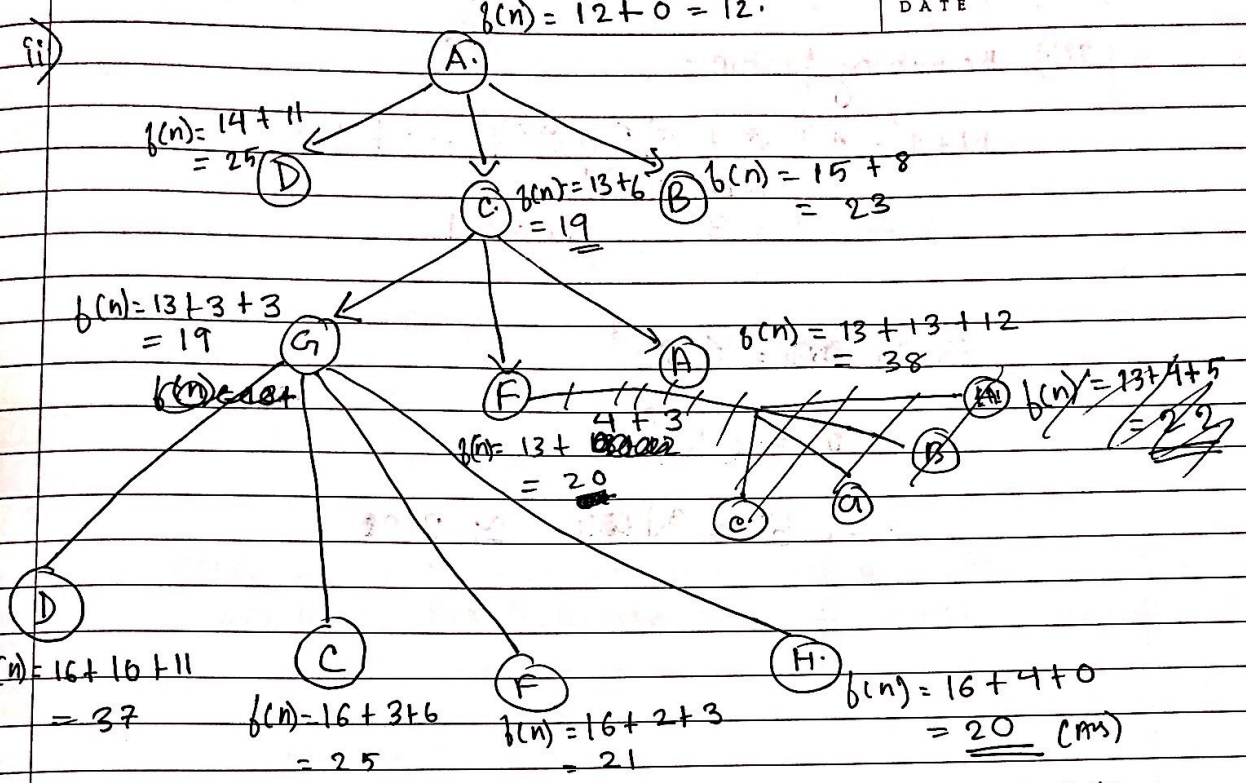
∴ pathcost > heuristic value

20 > 12.

(shown.)

DATE

ii)



~~If the cost of optimal solution be C*~~

let the optimal path cost be C*

Here, even after finding the solution, since there was a node with the same path cost as C*, it expanded the other node just to check if the ~~value~~ path cost to goal via that node is lesser than C*.

~~If there was a~~

Let the suboptimal Goal node be G,

$\therefore f(G) = g(G) + h(G)$

$f(G) > C^*$ (shown)

~~C* 0~~

~~22 > 20~~ (shown)

If there was a suboptimal path to the goal node, the A* search will not stop there, it will expand other nodes to find the most optimal solution.

Typo

DATE

(iii) Branching factor:-

$N+1 = 1 + b^* + (b^*)^2 + (b^*)^3 + \cdots + (b^*)^d$.

$N \approx (b^*)^d$

here, $d = 3$, $N = 11$.

$\Rightarrow 11 = (b^*)^3$

$\Rightarrow 11 = (b^*)^3$

$\therefore b^* = \sqrt[3]{11} \simeq 2.2$.

Question 4

DATE

ii)

String	n	x	F
010110	22	88	7744 x
100001	33	132	17424 x
010001	17	68	4624 x
101010	42	168	28224 x

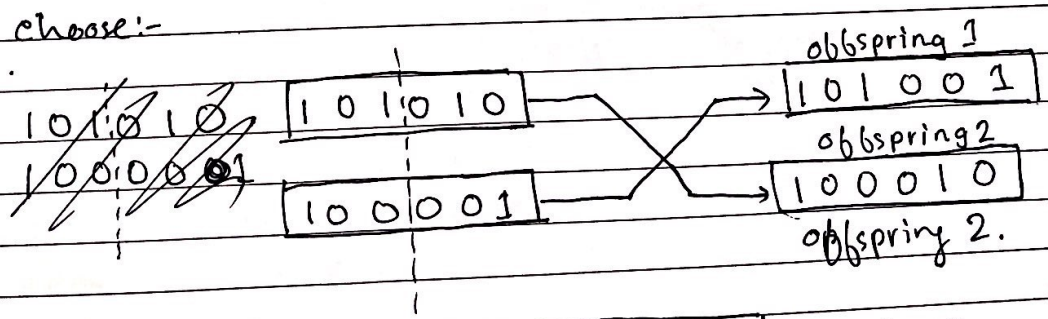
i) Here, we are always keeping k = 4 states in the memory where all of these works as a candidate for crossover :-

010110
100001
010001
101010

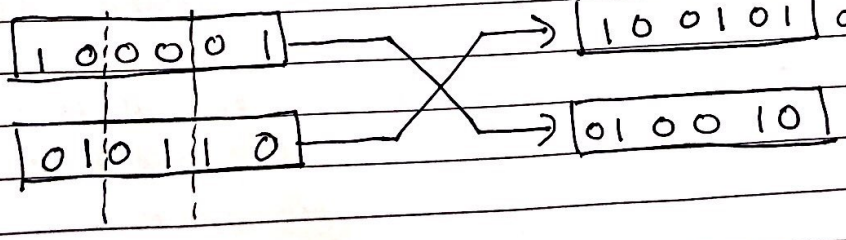
iii) The chromosomes chosen according to the fitness are :-

010110
100001
101010

iv) a.) we choose :-



b.)



Typo

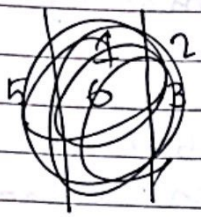
DATE

iv) Mutation:-

		Final Offspring.
101001	→	101011
100010	→	110010
100101	→	100101
010010	→	0100110

Question 5

DATE



Goal State.

1	2	3
4	5	6
7	8	

$f = 0$

Lets consider the manhattan distance as the heuristic value (since it dominates misplaced tiles)

let, $f = -$(manhattan distance)

start.

	1	2
5	6	3
4	7	8

$f = -(1+1+1+1+1+1+1+1)$

$= -(8)$

move up

5	1	2
	6	3
4	7	8

$f = -9$

Ridges.

move left.

1		2
5	6	3
4	7	8

$f = -7$

Local Maxima

move up

1	6	2
5		3
4	7	8

$f = -8$

Ridges

move left

1	2	
5	6	3
4	7	8

$f = -6$

Ridges

Two problemes that can occur are :-

① Local Maxima :- Hill climbing will get stuck at a local maxima, which is not a statisfactory solution.

② Ridges :- It will not overcome ridges. Search will only oscillate slowly.

Typo

DATE

To overcome these problems, we can modify the Hill climbing search:-

(1.) We can modify the choice of neighbour.
for example neighbours can be choosen randomly which will help Hill climbing search escape the ridges.

(2.) We can modify the criteria for accepting neighbour for the current state.
for example:- Accept the neighbour if it is better, if it is not better, then accept it with some fixed probability.
This helps HC to escape local maxima.