DATE

Theory

Q1 a. No the new algorithm will not guarantee to find the ~~opti~~ optimal Solution. Because in A* algorithm explored set play a vital role ~~in~~ to ensure that algorithm don't revisit the nodes that it had already processed. As the node already been processed the ~~algo~~ algorithm has already found the shortest path. Now if we remove this the algorithm will process multiple times each with different-different path length.

Now even if heuristic is admissible means it never reach the goal ~~Now~~ or find an optimal path. Now without explored set the algorithm ~~can~~ might end up selecting path that ~~was~~ was processed later. Hence without explored set it doesn't ~~gu~~ guarantee optimal solution.

b. Yes, the algorithm will still complete. Even without the explored set it will still continue to explore all paths until it finds the goal. It will not stop ~~at~~ until it has found a solution as ~~explo~~ explored all possible path.

DATE

c. ~~The new algorithm will be faster if it has explored set~~

c. The new algorithm will be faster if it has explored set as it ensures the node is processed only once.

Now without the explored set it may be faster in terms of memory usage and space complexity. But it could ~~cause~~ end up processing all the nodes even the same one ~~as~~ as well with different paths. This increases computational overhead of the algorithm & thus it slows down.

DATE

Q2 a. Using A^* Star Search

State

State	S	A	B	C	D	E	F	G	H	I	J	4
h(n)	1	8	6	2	7	3	2	0	9	11	13	4

As we all Know $f(n) = g(n) + h(n)$ ~~$g(n)+h(n)$~~

1) S0 $S \to A$, $S \to B$, $S - C$

$f(n) = 4+8$
$= 12$

$f(n) = 18+6$
$= 24$

, $f(n) = 11+2$
$= 13$

2) $S \to A \to D$ $f(n) = 4+5+7 = 16$

3) $S - A - D - H$ $f(n) = 19$

$S - A - D - I$ $f(n) = 19+11 = 30$

$S - A - D - F$ = ($f(n)$ = 12

4) $f(n) = S - A - D - F - G$

$f(n) = 10+13 = 23$

DATE

Now here G is the priority queue to here
to using best backtracking backtracking
of the lowest f(n)

So

S – A – D – H f(n) = 10+1+9 = 19

S – A – D – H – f(n) = 11+11 = 22

S – A – D – H – I f(n) = 12+13 = 25

S – A – D – H – I – G f(n) = 14

So path found ~~path cost~~ path cost S – A – D – H – I – G

f(n) = 14

Best first search

from table value path for h(n)

(1) Initialising open [A, B, C], ~~close~~ close(S)

(2) Opening [A, B, D, E, F], Closing ~~[S~~ [S, C]

(3) Opening [A, B, D, E, G], closing [S, C, F]

(4) Opening [A, B, D, E], Closing [S, C, F, G]

Here S – C – F – G f(n) = 11+2+13 = 26

path cost = 26

C. Using by Dijkstra Algorithm

From table value /

Tent Vertex	S	A	B	C	D	E	F	J	G	H	I	
S	0	0	0	0	0	0	0	0	0	0	0	0
A	∞	4	4	4	4	4	4	4	4	4	4	4
B	∞	18	8	8	8	8	8	8	8	8	8	8
C	∞	11	11	11	11	11	11	11	11	11	11	11
D	∞	∞	5	∞	13	13	13	13	13	13	13	13
E	∞	∞	∞	∞	20	20	20	20	20	20	20	20
F	∞	∞	∞	∞	2	1	1	1	1	1	1	1
G	∞	∞	∞	∞	8	∞	19	13	3	3	3	3
H	∞	∞	∞	∞	∞	1	1	1	1	1	1	1
I	∞	∞	∞	∞	∞	20	20	20	20	30	20	2
J	∞	∞	∞	∞	∞	∞	∞	∞	5	2	2	1
K	∞	∞	∞	∞	∞	∞	∞	∞	13	7	7	7

Path S — A — D — H — Z — 9

4 5 1 1 3

Cost ~~Adding the cost~~ is 14

~~Ao Ao~~

~~Ao~~ Adding all these weights

14