AI Assignment 2 (Theory)

Akash Kushwaha

2021514

(a) If the heuristic is admissible then the new algorithm will still find the optimal solution without keeping the explored set, the algorithm will re-explore some states, but the admissible nature of the heuristic ensures that the new path to the goal will never have a lower actual cost than the one previously found.
Also, a node that is already expanded (relaxed) then it will not again be relaxed if discovered again through another node so finally algorithm will find the optimal solution and won't get stuck in an infinite loop.

(b) Completeness means the algorithm will surely find a solution if one exists and if doesn't exist it should report failure. So according to the discussion in (a) we can clearly infer that if the solution exists then the algorithm would find it surely but if it doesn't exist then while running the algorithm at one step all nodes would be relaxed and the heap will become empty then the algorithm terminates which indicates that algorithm is complete.

(c) Although it doesn't get stuck into an infinite loop it will visit the same node multiple times due to the absence of an explored set and hence algorithm will not be faster rather it will become inefficient and slower due to redundant computations.

Q2.

A* Search:-

S→C : f(n)= 11+2=13

S→B : f(n)= 18+6= 24

S→A : f(n)= 4+8=12

S→A→B : f(n)= 12+6=18

S→A→D : f(n)= 9+7 = 16

S→C→D : f(n)= 24+7 = 31

S→C→F : f(n)= 13+2= 15

S→C→ E : f(n)= 31+3 = 34

S→C→F→G : f(n)= 26+0 = 26

S→A→D→H : f(n)= 10+9 =19

S→A→D→F : f(n)= 10+2=12

S→A→D→I : f(n)= 29+11= 40

S→A→D→F→G : f(n)= 23+0 = 23

S→A→B → (No further path Towards Goal)

S→A→D→H→I : f(n)= 11+11=22

S→A→D→H→J : f(n)= 12+13= 25

S→A→D→H→I→G : f(n)=14+0 = 14 ← optimal cost

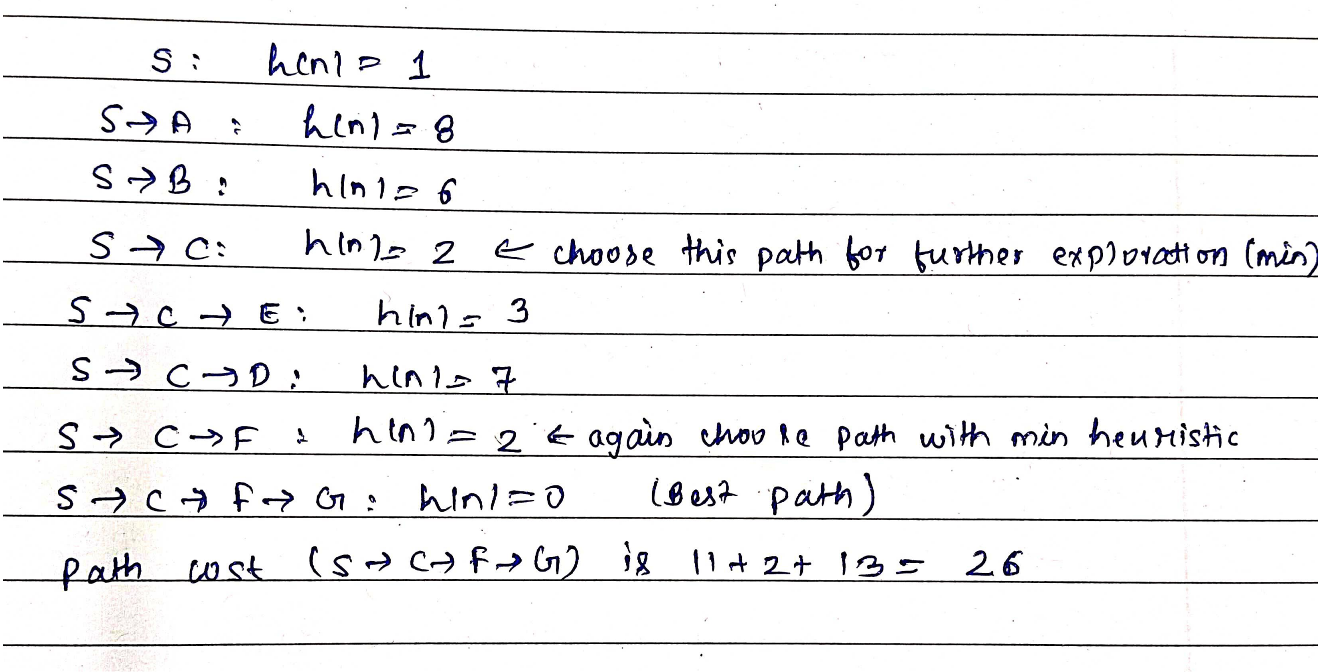
S→A→D→H→I→J : f(n)= 16+13 = 29

S→A→D→H→I→K : f(n)= 23+4 =27

Hence, Optimal cost from S to G is 14 using A*

BFS (Best First Search):-

Best first search pops out the node with minimum heuristic value from the heap to expand further.



S : h(n) = 1

S → A : h(n) = 8

S → B : h(n) = 6

S → C : h(n) = 2 ← choose this path for further exploration (min)

S → C → E : h(n) = 3

S → C → D : h(n) = 7

S → C → F : h(n) = 2 ← again choose path with min heuristic

S → C → F → G : h(n) = 0 (Best path)

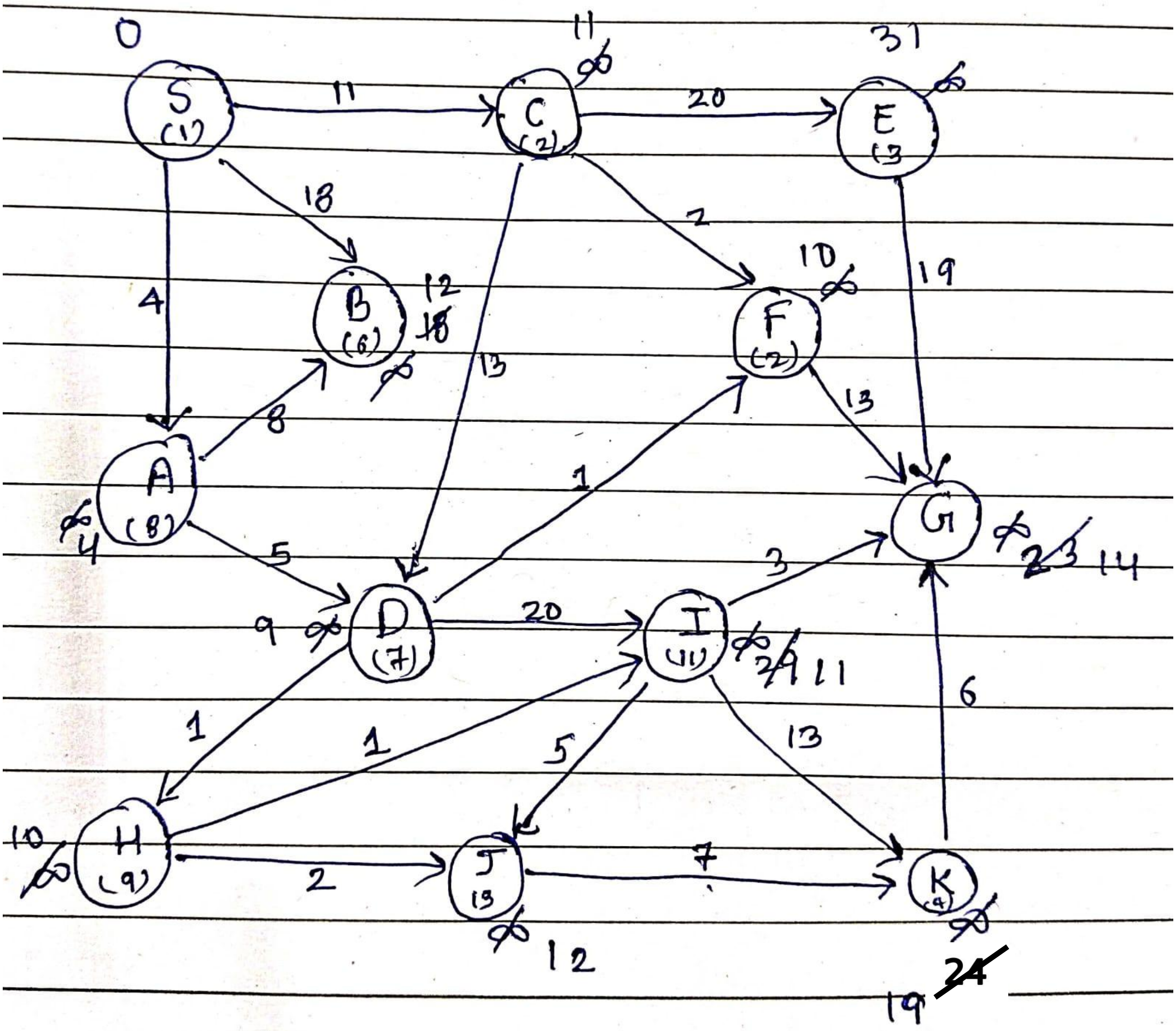
path cost (S → C → F → G) is 11 + 2 + 13 = 26

So, final path is S-->C-->F-->G and Cost is 26.

Dijkstra:-

```
Final Shortest path from S to G is:
S-->A-->D-->H-->I-->G and Cost is 14.
Shortest distances from node S to all other nodes:
Cost from S to Node S is: 0
Cost from S to Node C is: 11
Cost from S to Node E is: 31
Cost from S to Node K is: 19
Cost from S to Node I is: 11
Cost from S to Node J is: 12
Cost from S to Node H is: 10
Cost from S to Node D is: 9
Cost from S to Node A is: 4
Cost from S to Node F is: 10
Cost from S to Node B is: 12
Cost from S to Node G is: 14
```

S	A	B	C	D	E	F	G	H	I	J	K
~~0~~	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
~~0~~	~~4~~	18	11	∞	∞	∞	∞	∞	∞	∞	∞
~~0~~	~~4~~	12	11	~~9~~	∞	∞	∞	∞	∞	∞	∞
~~0~~	~~4~~	12	11	~~9~~	∞	~~10~~	∞	10	29	∞	∞
~~0~~	~~4~~	12	11	~~9~~	∞	~~10~~	23	~~10~~	29	∞	∞
~~0~~	~~4~~	12	~~11~~	~~9~~	∞	~~10~~	23	~~10~~	11	12	∞
~~0~~	~~4~~	12	~~11~~	~~9~~	31	~~10~~	23	~~10~~	~~11~~	12	∞
~~0~~	~~4~~	~~12~~	~~11~~	~~9~~	31	~~10~~	14	~~10~~	~~11~~	~~12~~	24
~~0~~	~~4~~	~~12~~	~~11~~	~~9~~	~~31~~	~~10~~	~~14~~	~~10~~	~~11~~	~~12~~	~~19~~


0
S
(1)
11
C
(2)
20
31
E
(3)
18
4
B
(6)
12
18
∞
13
2
10
∞
F
(2)
19
8
A
(8)
∞
4
5
1
13
G
∞
23
14
3
9
∞
D
(7)
20
I
(11)
∞
29
11
1
1
5
13
6
10
∞
H
(9)
2
J
(13)
∞
12
7
K
(4)
∞
24
19