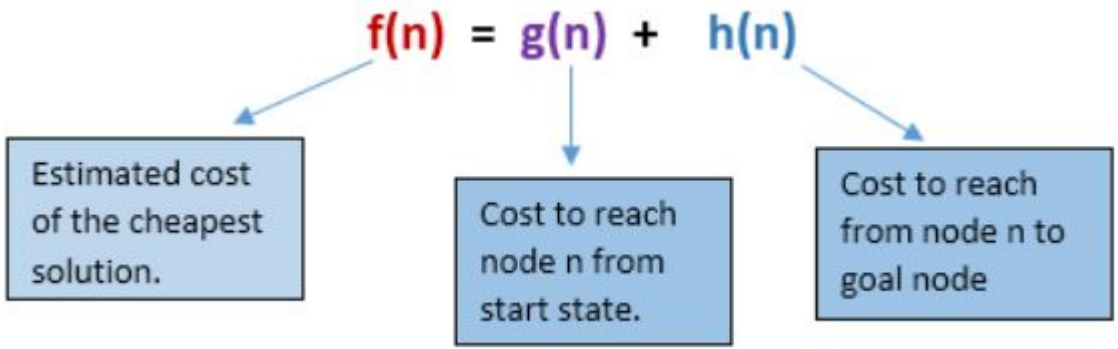
A* Search

- It uses heuristic function h(n), and cost to reach the node n from the start state g(n).
- A* search algorithm finds the shortest path through the search space using the heuristic function.
- This search algorithm expands less search tree and provides optimal result faster.



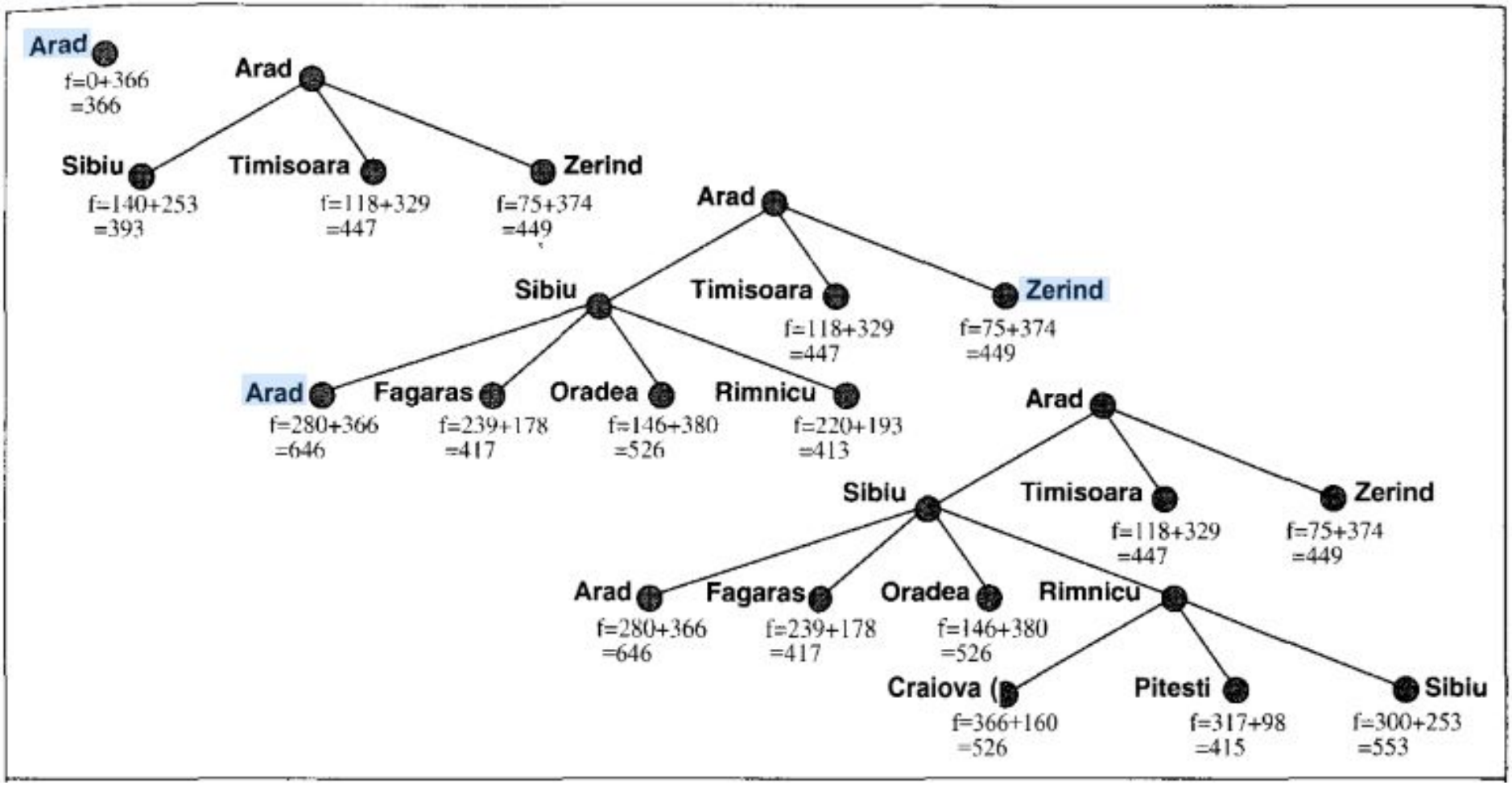
A* Search

Advantages:

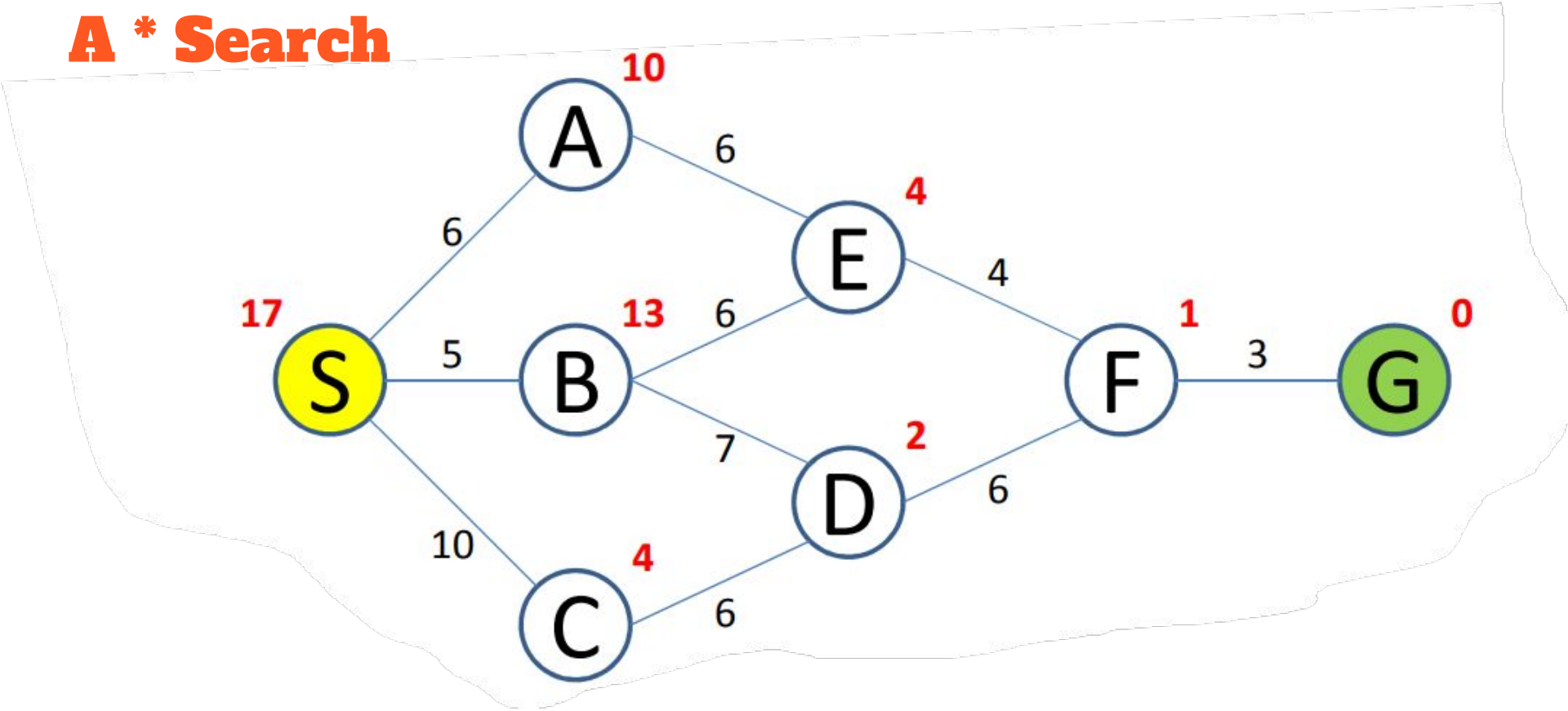
- A* search algorithm is the best algorithm than other search algorithms.
- A* search algorithm is optimal and complete.
- This algorithm can solve very complex problems.

Disadvantages:

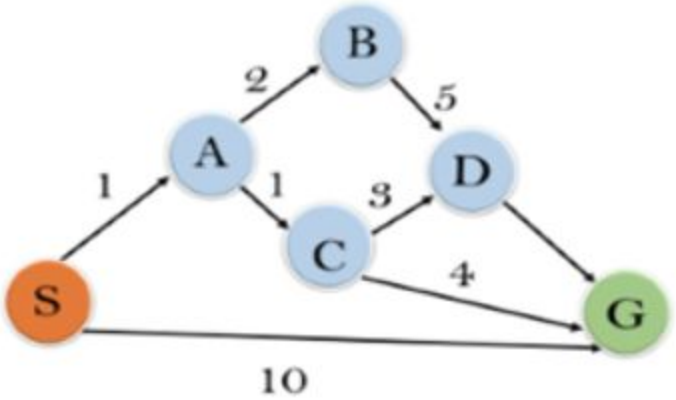
- It does not always produce the shortest path as it mostly based on heuristics and approximation.
- A* search algorithm has some complexity issues.
- The main drawback of A* is memory requirement as it keeps all generated nodes in the memory, so it is not practical for various large-scale problems.


Arad
f=0+366
=366
Arad
Sibiu
f=140+253
=393
Timisoara
f=118+329
=447
Zerind
f=75+374
=449
Arad
Sibiu
Timisoara
f=118+329
=447
Zerind
f=75+374
=449
Arad
f=280+366
=646
Fagaras
f=239+178
=417
Oradea
f=146+380
=526
Rimnicu
f=220+193
=413
Arad
Sibiu
Timisoara
f=118+329
=447
Zerind
f=75+374
=449
Arad
f=280+366
=646
Fagaras
f=239+178
=417
Oradea
f=146+380
=526
Rimnicu
Craiova (
f=366+160
=526
Pitesti
f=317+98
=415
Sibiu
f=300+253
=553

A * Search

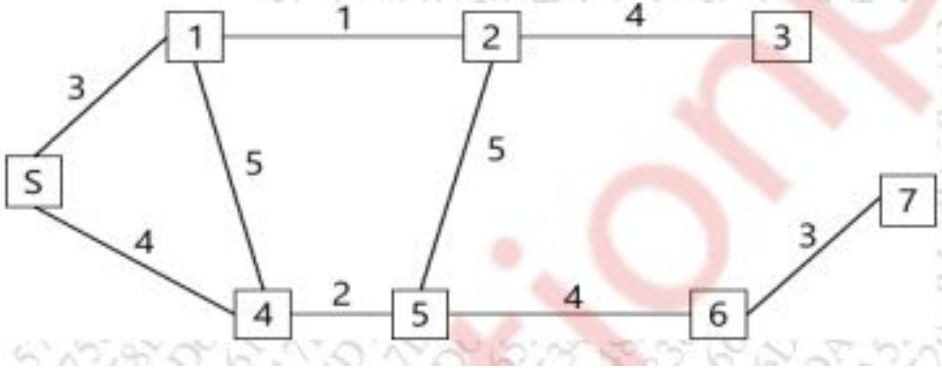


A * Search



State	h(n)
S	5
A	3
B	4
C	2
D	6
G	0

Consider the graph given in Figure 1 below. Assume that the-initial state is S and the goal state is 7. Find a path from the initial state to the goal state using A* Search. Also report the solution cost. The straight line distance heuristic estimates for the nodes are as follows: h(1)=14, h(2)=10, h(3)=8, h(4)=12, h(5)=10, h(6)=10, h(S)=15. **[10]**



(b) Consider the graph given in Figure 2 below. Assume that the initial state is A and the goal state is G. Show how **Greedy Best first Search** would create a search tree to find a path from the initial state to the goal state:

[10]

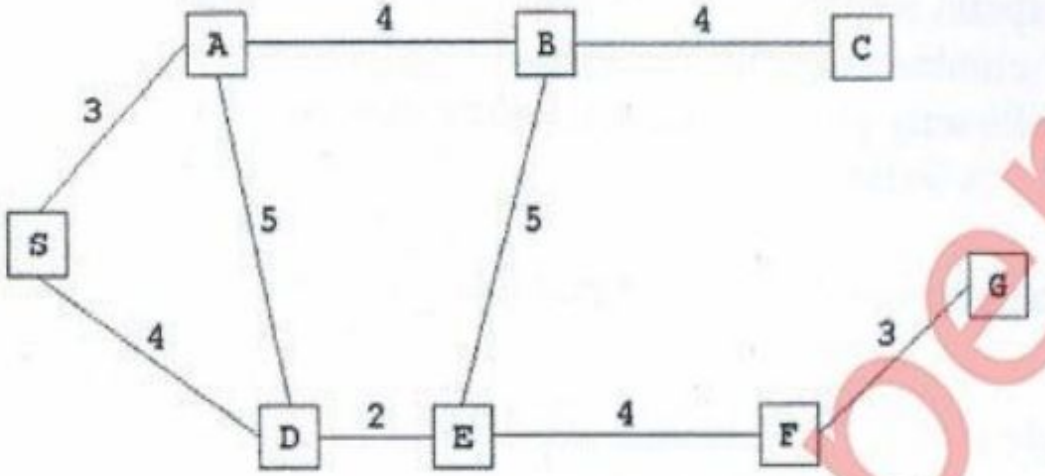


Figure 2.

At each step of the search algorithm, show which node is being expanded, and the content of fringe. Also report the eventual solution found by the algorithm, and the solution cost. Assuming the straight-line distance as the heuristics function: h(S)=10.5, h(A)=10, h(B)=6, h(C)=4, h(D)=8, h(E)=6.5, h(F)=3 and h(G)=0.

(b) Consider the graph given in Figure 2 below. Assume that the initial state is **S** and the goal state is **G**. Show how **A* Search** would create a search tree to find a path from the initial state to the goal state: [10]

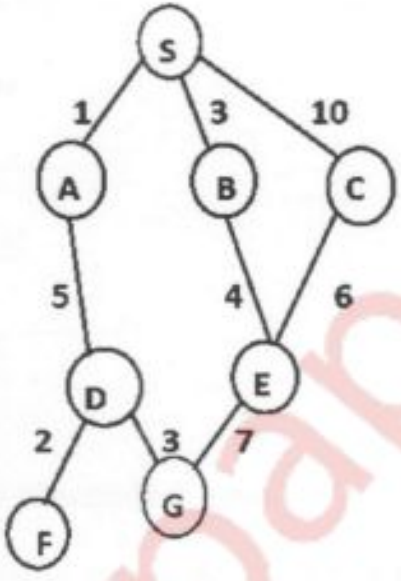
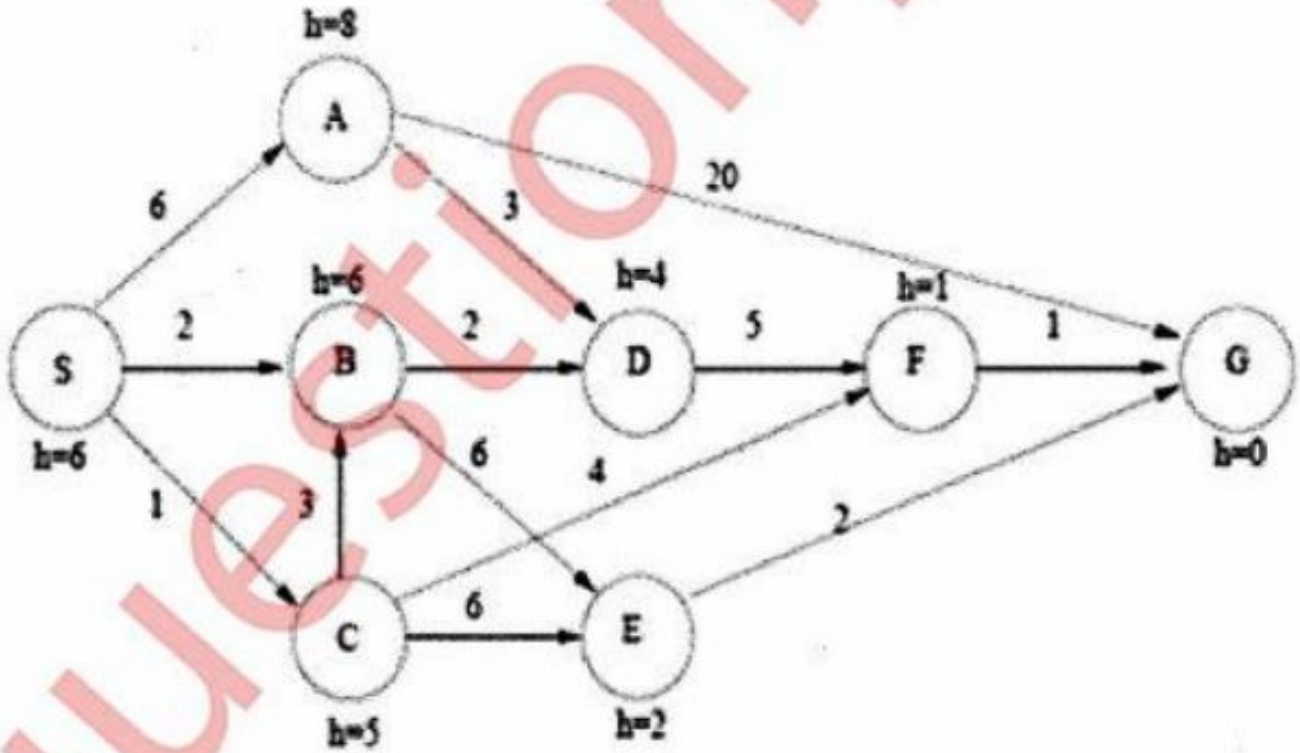
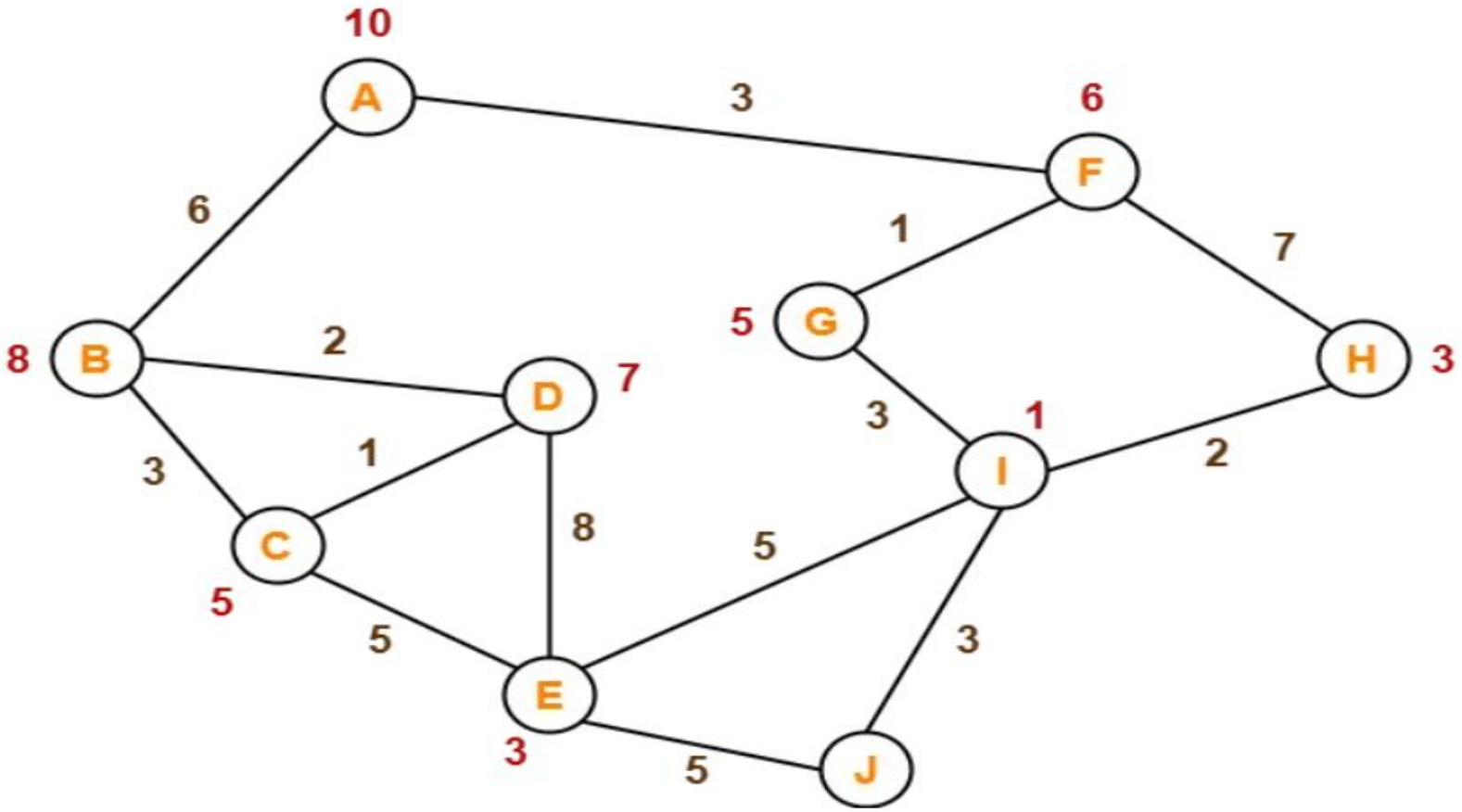


Figure 2

At each step of the search algorithm, show which node is being expanded, and the content of fringe (OPEN). Also report the eventual solution found by the algorithm, and the solution cost. Assuming the straight-line distance as the heuristics function: h(S)= 13, h(A)=7, h(B)=9, h(C)=11, h(D)=2, h(E)=4, h(F)=1, and h(G)=0.

Consider the search problem below with start state S and goal state G. The transition costs are next to the edges and the heuristic values are next to the states. What is the final cost using A * search. [10]




10
A
3
6
F
6
1
7
5
G
8
B
2
H
3
D
7
3
1
1
I
2
3
8
5
C
5
5
E
3
3
5
J