1-Suppose you have the following search space:

a) Draw the state space of this problem.

b) Assume that the initial state is A and the goal state is G. Show how each of the following search strategies would create a search tree to find a path from the initial state to the goal state. **(Assume that repeated nodes are not possible)**

- Uniform cost search (UCS)
- Greedy search
- A* search

c) 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 each algorithm, and the solution cost.

State	next	cost
A	B	4
A	C	1
B	D	3
B	E	8
C	C	0
C	D	2
C	F	6
D	C	2
D	E	4
E	G	2
F	G	8

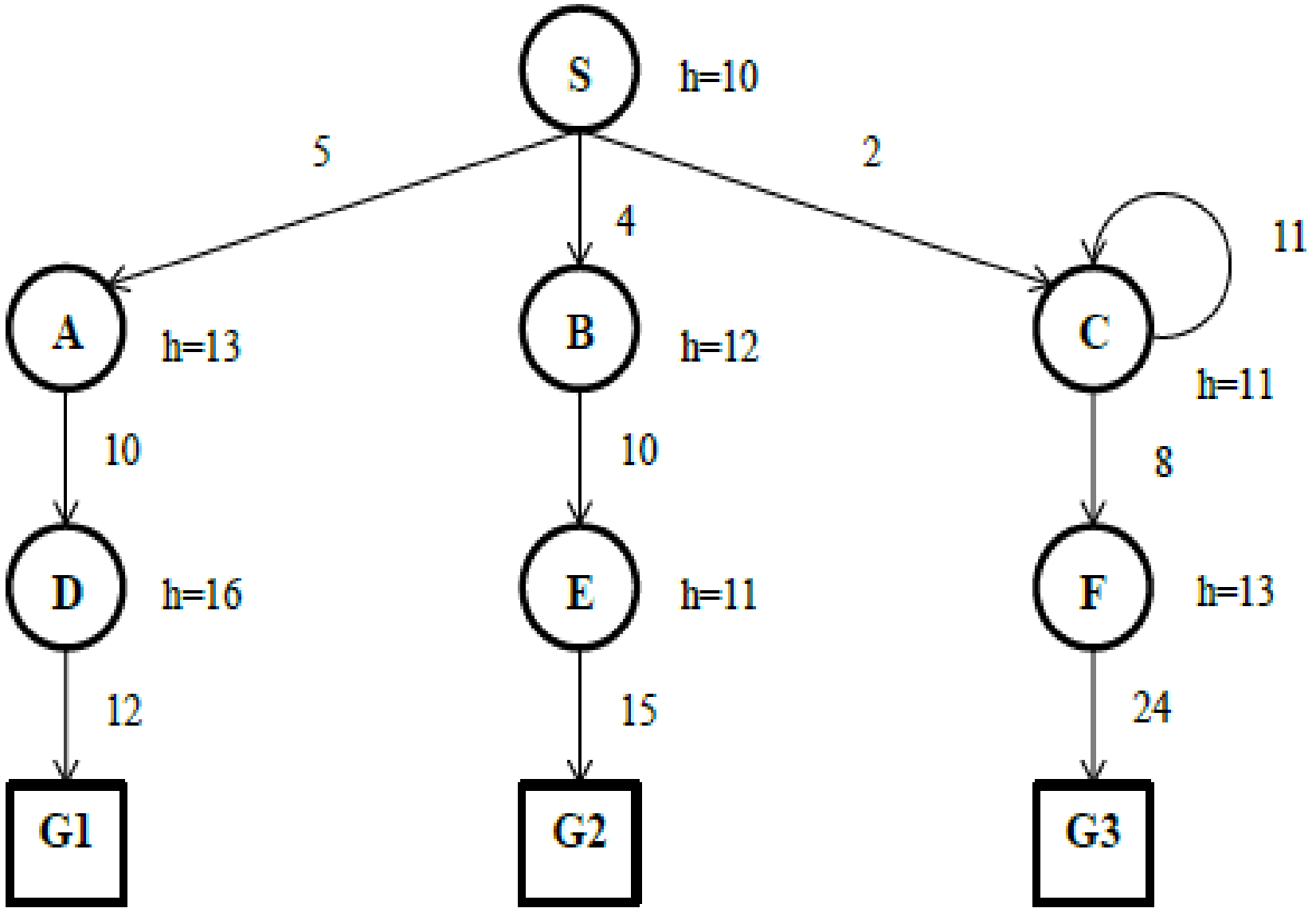
state	h
A	8
B	8
C	6
D	5
E	1
F	4
G	0

State	next	cost
A	B	4
A	C	1
B	D	3
B	E	8
C	C	0
C	D	2
C	F	6
D	C	2
D	E	4
E	G	2
F	G	8

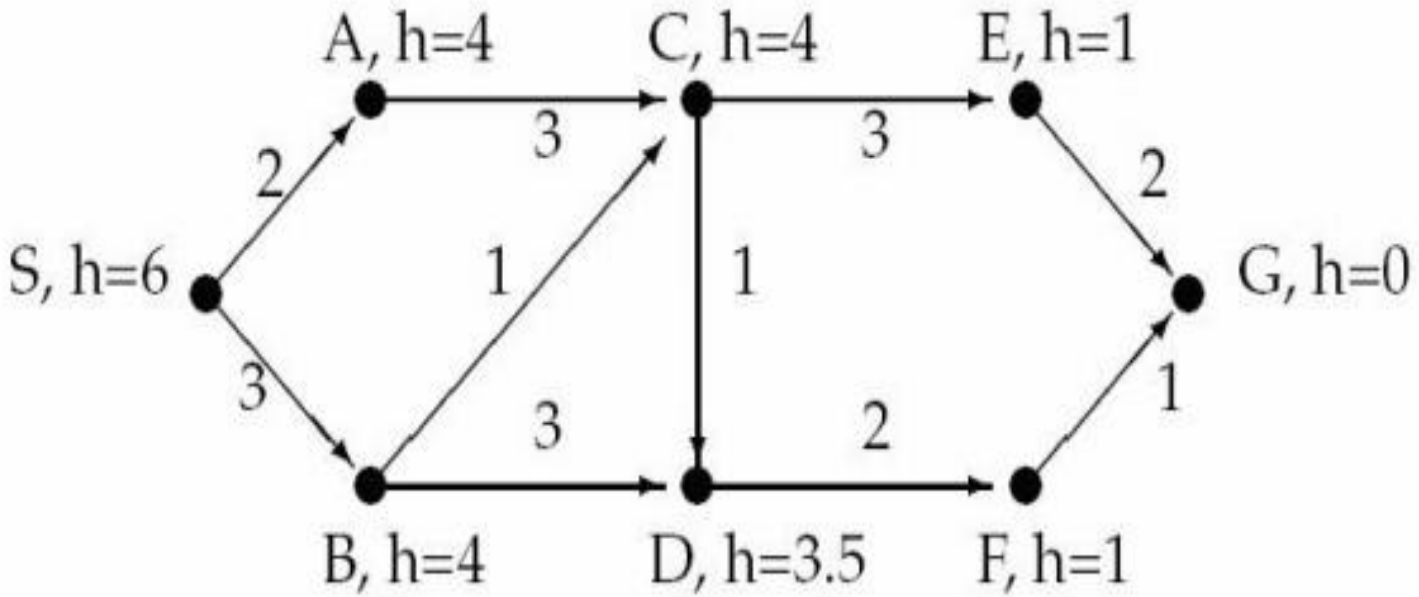
state	h
A	8
B	8
C	6
D	5
E	1
F	4
G	0

2-Execute Tree Search through the given graph (do not remember visited nodes, so repeated nodes are possible).
Step costs are given next to each arc, and heuristic values are given next to each node (as h=x). The successors of each node are indicated by the arrows out of that node. **(Note: C is a successor of itself).**
For each search strategy below, indicate the order in which nodes are expanded, ending with the goal node that is found.

- Uniform Cost Search
- Greedy Best-First Search
- A* Search


S
h=10
5
4
2
11
A
h=13
B
h=12
C
h=11
10
10
8
D
h=16
E
h=11
F
h=13
12
15
24
G1
G2
G3

3- Suppose we want to use the A* algorithm on the graph given to find the shortest path from node S to node G. Each node is labeled by a capital letter and the value of a heuristic function. Each edge is labeled by the cost to traverse that edge.



- For this problem:
- Perform the A* algorithm on this graph, filling in the table below. You should not need all the lines in the table. Indicate the f, g, and h values of each node on the queue as shown in the first two rows of the table.

Iteration	Node expanded	Priority queue at the end of this iteration
0		S = 0 + 6 = 6 (f(S) = g(S) + h(S))

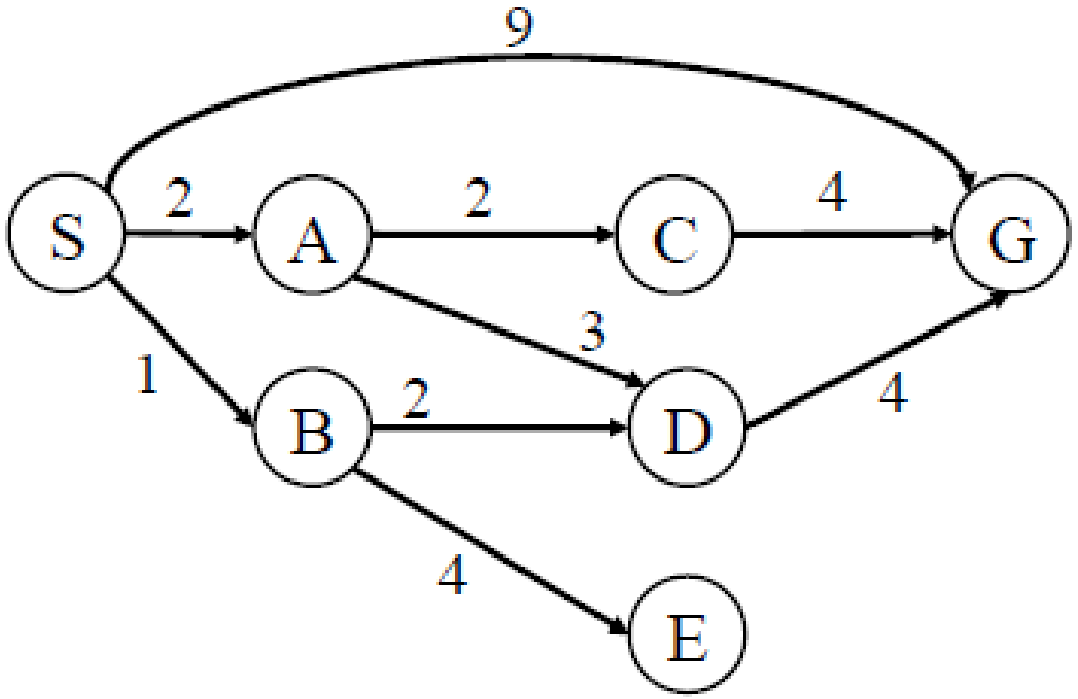
- You need not write the contents of the (priority) queue in order in the table.
- Assume that if you find a path to a node already on the queue that you update its cost (using the lower f value) instead of adding another copy of that node to the queue.
- Show the path found by the A* algorithm on the graph above

- Is it the optimal path? Explain your answer in terms of **admissibility** and **consistency**.

- If it is not the optimal solution, how can you make it optimal?

4- The following search problem, represented as a graph is given. The start state is S and the only goal state is G.

The given problems reference both tree search and graph search. For questions which require a heuristic, use the one given below.



Heuristic

S	A	B	C	D	E	G
6	0	6	4	1	10	0

3-

a) What path will UCS tree search return?

b) What path will UCS graph search return?

c) What path will greedy tree search return?

d) What path will A* ***tree*** search return?

e) What path will A* ***graph*** search return?

f) Explain why your answer to part (e) is reasonable.