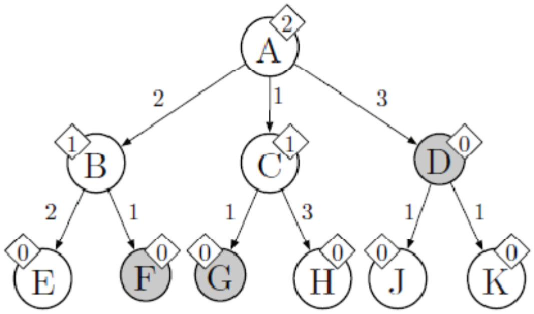
Registration No: -------------------------   Section: -------------

Question 2. [Greedy , UCS, A*]

Consider the state space search problem shown to the right. **A** is the **start state** and the **shaded states** are **goals**. Arrows encode possible state transitions, and numbers by the arrows represent action costs. Note that state transitions are directed; for example, **A -> B** is a valid transition, but **B -> A** is not. Numbers shown in diamonds are heuristic values that estimate the optimal (minimal) cost from that node to a goal.



For each of the following search algorithms, write down the nodes that are removed from fringe in the course of the search, as well as the final path returned.

Because the original problem graph is a tree, the tree and graph versions of these algorithms will do the same thing, and you can use either version of the algorithms to compute your answer.

Assume that the data structure implementations and successor state orderings are all such that ties are broken alphabetically. For example, a partial plan **S -> X -> A** would be expanded before **S -> X -> B;** similarly, **S -> A -> Z** would be expanded before **S -> B -> A.**

(a) Greedy-Best-First Search **[2 Points]**

Nodes removed from fringe:

Path returned:

(b) Uniform-Cost Search **[2 Points]**

Nodes removed from fringe:

Path returned:

(c) A* Search **[3 Points]**

Nodes removed from fringe:

Path returned: