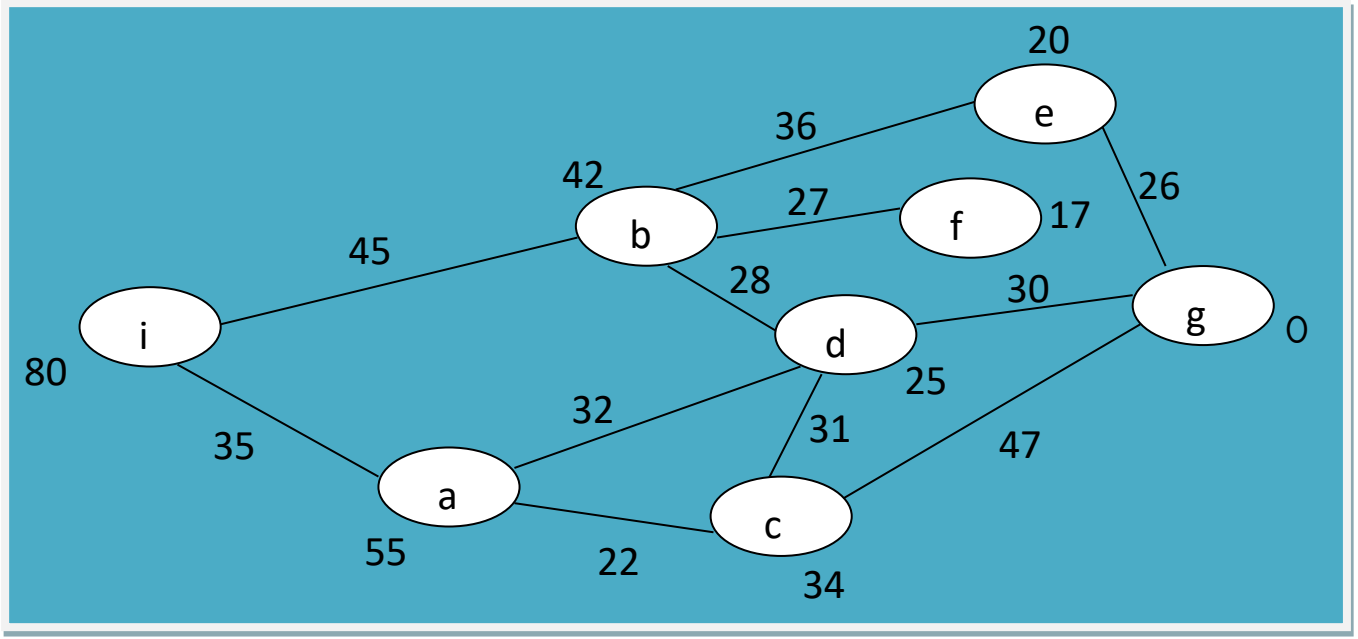
3.2. Greedy Best-First Search

- An example (problem instance):



i – Initial state/node g – Goal state/node

Tabular representation of the graph:

Node	Neighbor	Distance
i	a	35
i	b	45
a	c	22
…	…	…

Node	h(Node)
i	80
a	55
b	42
…	…

Execution of the algorithm: …….

❖ **Important characteristics of the algorithm:**

1. It is an instance of a general graph search.
2. It is a heuristic search.
3. A node is selected for expansion based on an evaluation function, f(n), which is taken

   f(n) = h(n),

   that means, expand the node which 'is closest to the goal according to the heuristic function' or 'seems best'.
4. A Priority Queue (PQ), which contains nodes in ascending order of h-values, is also maintained. The PQ offers the node for expansion.
5. A Possible Path (PP) is maintained that contains nodes currently supposed to be in the solution.
6. A tree of visited nodes along with their children is also maintained which helps to update PQ and PP.

7. The process begins by placing the source node (initial state) in the empty PQ, and initiating a tree by placing that node as its root.

   The process terminates when the destination node (goal state) is placed in the PQ, and selected, consequently.

8. The 1st node from the PQ is selected repeatedly, and each time the tree, the PQ and the PP are updated:

   ✓ The node in the tree is marked visited and its neighbors from the graph are added to the tree as its children, while no repeated node is allowed in the tree;

   ✓ The node itself is deleted from the PQ, but its children, added to the tree are also added to the PQ.

   ✓ The PP is straightened up to the root from the selected node.

9. The strategy is 'greedy', may return a non-optimal solution.

10. The strategy is also not complete in the sense that it may unreasonably lead to great depth.