INFORMED SEARCH STRATEGIES
(Part-1)

Er. Zeeshan Ali Siddiqui
Assistant Professor
Deptt. of C.S.E.

FACULTY OF ENGINEERING AND TECHNOLOGY
UNIVERSITY OF LUCKNOW
LUCKNOW

INFORMED SEARCH STRATEGIES

Informed Search

- Informed search algorithm contains an array of *knowledge* such as how far we are from the goal, path cost, how to reach to goal node, etc.

- This knowledge help agents to explore less to the search space and find more *efficiently* the goal node.

- The informed search algorithm is more *useful* for large search space.

- Informed search algorithm uses the idea of *heuristic*, so it is also called Heuristic search.

- *A heuristic technique, or a heuristic, is any approach to problem solving or self-discovery that employs a practical method that is not guaranteed to be optimal, perfect, or rational, but is nevertheless sufficient for reaching an immediate, short-term goal or approximation.* *~Wikipedia*

Heuristics Function 1/2

- Heuristic is a function which is used in Informed Search, and it finds the most *promising* path.
- It takes the current state of the agent as its input and produces the estimation of how close agent is from the *goal*.
- The heuristic method, however, might not always give the best solution, but it guaranteed to find a good solution in *reasonable* time.
- Heuristic function *estimates* how close a state is to the goal.

Heuristics Function 2/2

- Heuristic function is represented by h(n), and it calculates the cost of an *optimal* path between the pair of states.

- The value of the heuristic function is always *positive*.

- Heuristic function is given as:

$$h(n) <= h^*(n)$$

Here,

- **h(n)** is heuristic cost, and
- **h*(n)** is the estimated cost.

- Hence, heuristic *cost* should be less than or equal to the estimated cost.

BEST FIRST SEARCH

Best First Search

- Idea: use an evaluation function f(n) for each node
  - f(n) provides an estimate for the total cost.
  - Expand the node n with smallest f(n).

- Implementation:
- Order the nodes in fringe increasing order of cost.

- Special cases:
  - greedy best-first search
  - A* search

GREEDY BEST FIRST SEARCH

Greedy Best First Search 1/2

- Greedy best-first search algorithm always selects the path which appears *best* at that moment. It is the combination of DFS and BFS algorithms.

- In BFS and DFS, when we are at a node, we can consider any of the *adjacent* as next node.

- So both BFS and DFS *blindly* explore paths without considering any cost function.

- The idea of Best First Search is to use an evaluation function to decide which adjacent is most *promising* and then explore.

Greedy Best First Search 2/2

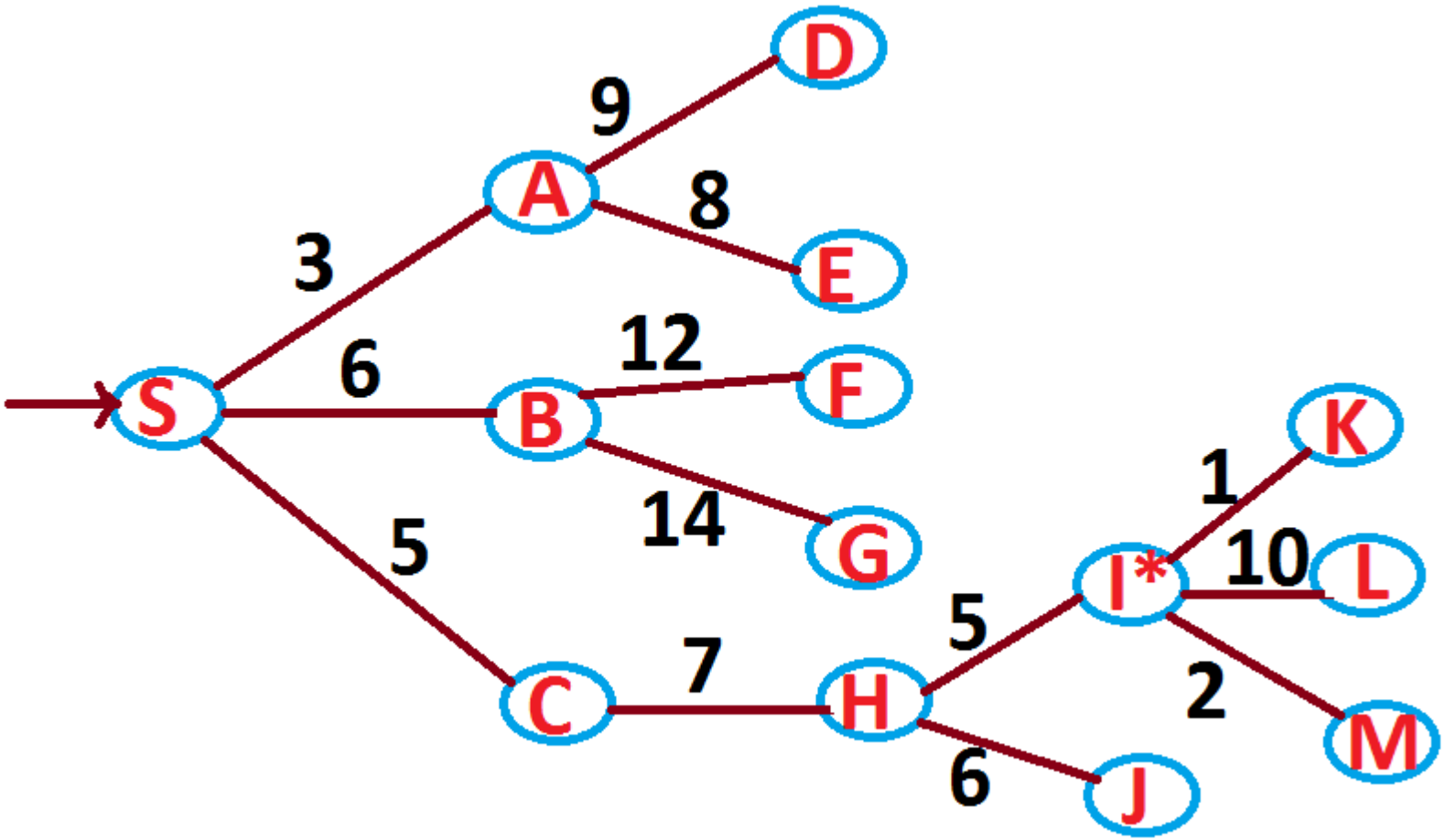
- Best-first search allows us to take the *advantages* of both algorithms.
- With the help of best-first search, at each step, we can choose the most *promising* node.
- In the best first search algorithm, we expand the node which is closest to the goal node and the closest cost is estimated by *heuristic function*, i.e.

$$f(n) = g(n)$$

- Were, h(n)= estimated cost from node n to the *goal*.
- We use a *priority queue* to store costs of nodes. So the implementation is a variation of BFS, we just need to change Queue to PriorityQueue.

Example 1/6

- We start from source "S" and search for goal "I" using given costs.

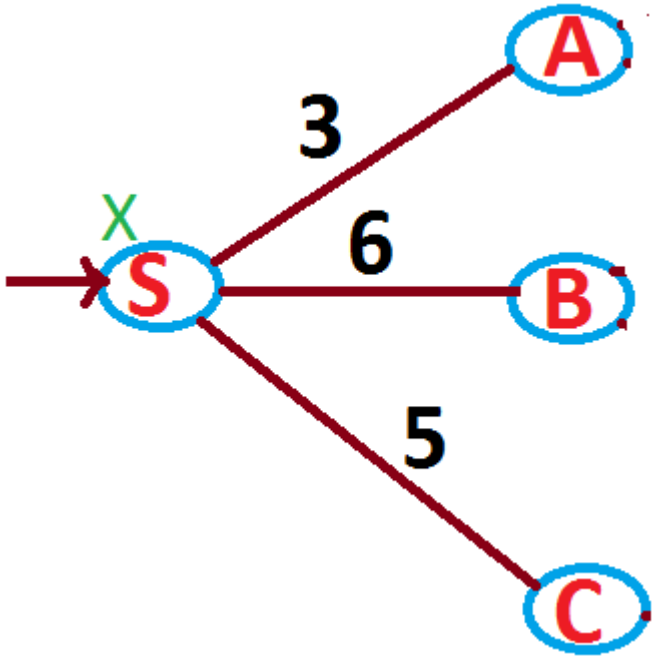


Example 2/6

- PriorityQueue initially contains S

S

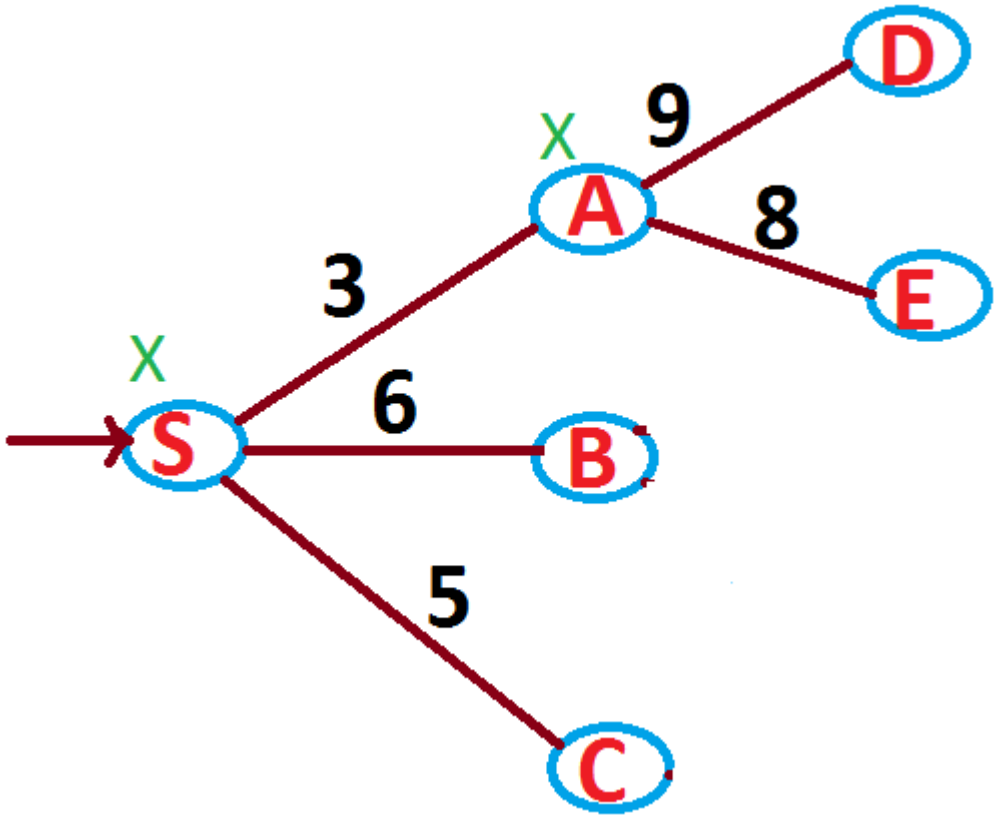
- Remove S from PriorityQueue and process unvisited neighbors of S to PriorityQueue
- PriorityQueue now contains {A, C, B} (C is put before B because C has lesser cost)



A	C	B

Example 3/6

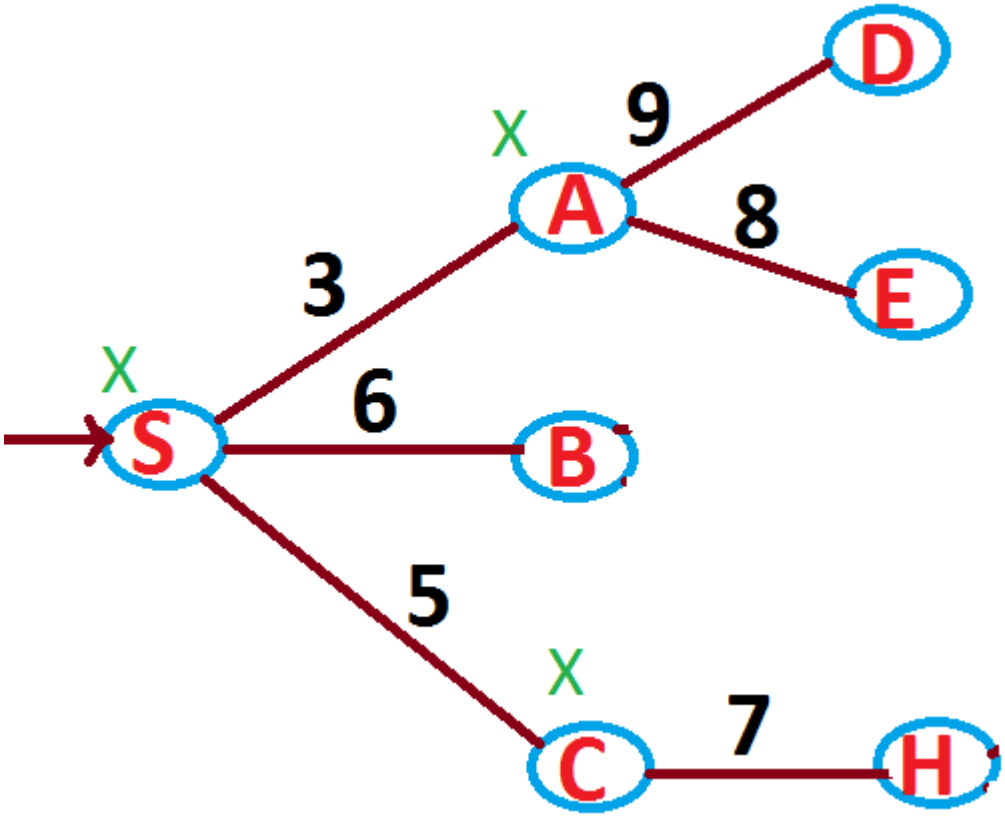
- Remove A from PriorityQueue and process unvisited neighbors of A to PriorityQueue.
- PriorityQueue now contains {C, B, E, D}



C	B	E	D

Example 4/6

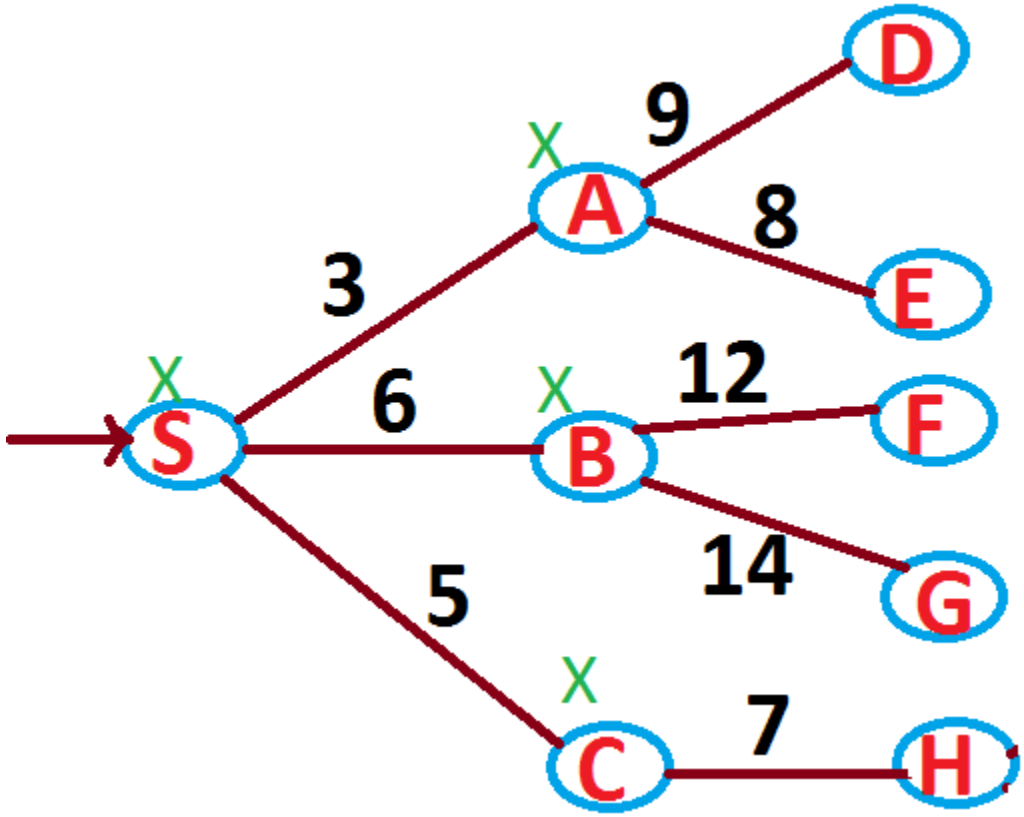
- Remove C from PriorityQueue and process unvisited neighbors of C to PriorityQueue.
- PriorityQueue now contains {B, H, E, D}



B	H	E	D

Example 5/6

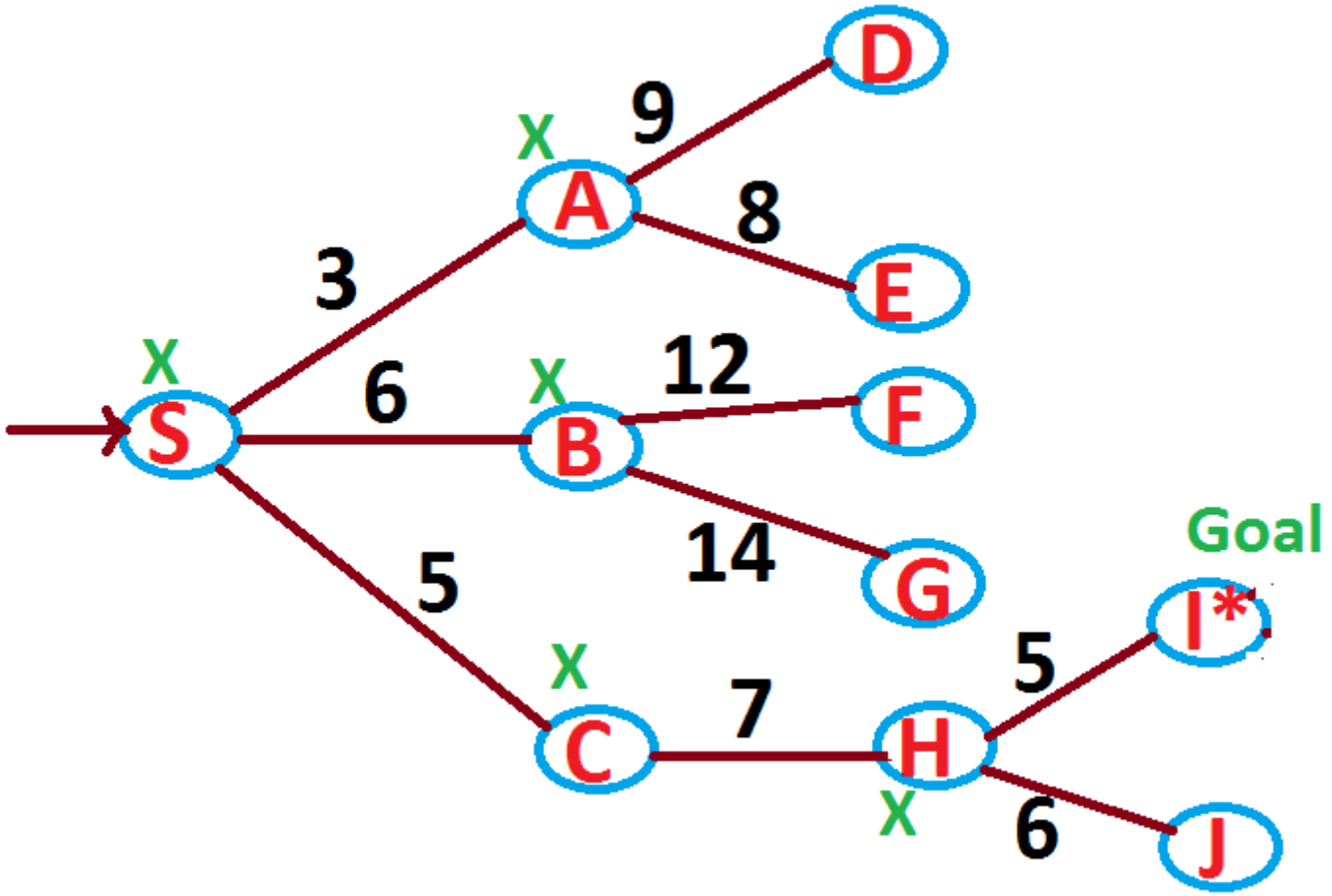
- Remove B from PriorityQueue and process unvisited neighbors of B to PriorityQueue.
- PriorityQueue now contains {H, E, D, F, G}



H	E	D	F	G

Example 6/6

- Remove H from PriorityQueue. Since our goal "I" is a neighbor of H, we return.



Takeaways

- Advantages:
  - Best first search can switch between BFS and DFS by gaining the advantages of both the algorithms.
  - This algorithm is more efficient than BFS and DFS algorithms.

- Disadvantages:
  - It can behave as an unguided depth-first search in the worst case scenario.
  - It can get stuck in a loop as DFS.
  - This algorithm is not optimal.

- Note: Performance of the algorithm depends on how well the cost or evaluation function is designed.

References

1. Stuart Russell,Peter Norvig, “Artificial Intelligence-A Modern Approach”, Pearson.
2. Elaine Rich and Kevin Knight, “Artificial Intelligence”, McGraw-Hill.
3. E Charniak and D McDermott, “Introduction to Artificial Intelligence”, Pearson.
4. https://www.javatpoint.com/ai-informed-search-algorithms
5. https://www.geeksforgeeks.org/best-first-search-informed-search/

Thank You.