Completeness	Yes	Yes
Optimality	Yes	No
Time-complexity	Consume more time	Consume less time
Space complexity	Consume more memory	Consume less memory

Tutorial 4

1. **Figure 1.1** below shows a Block World Problem. A robot will move the blocks one by one from initial state S1 to reach the goal state S19.

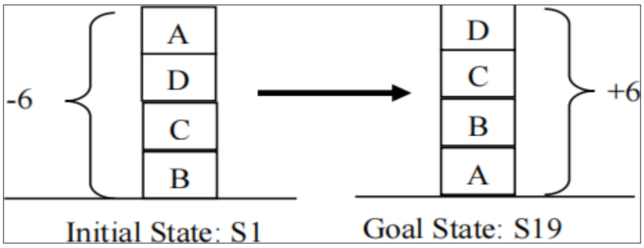


Figure 1.1

Figure 1.2 shows the state space of the Block World Problem and the heuristic costs for each state are shown in parentheses next to their respective nodes.

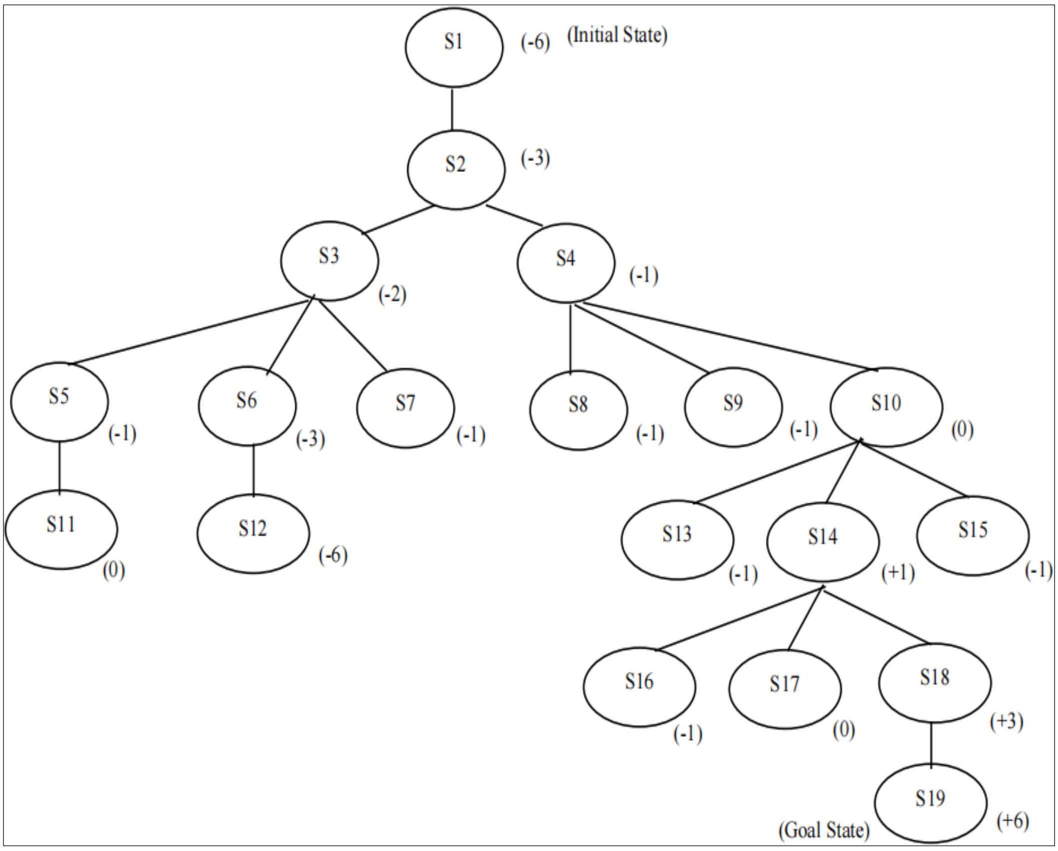


Figure 1.2

a) Explain step cost used in problem formulation. Specify the value of the step cost for the problem above.

b) Hill climbing search is unable to guarantee completeness and optimality as it may be trapped into local maximum.
   i. Explain local maximum.
   ii. Discuss why hill climbing search always lead to a local maximum.
   iii. Use simple hill-climbing and steepest-ascent hill-climbing to search for the best path from S1 to S19 on the state space shown in **Figure 1.2**. Then for each search technique, draw the resulting search tree that shows the visited nodes. Show that hill-climbing technique can be trapped into a local maximum.

c) A search technique can be evaluated based on four criteria: completeness, optimality, time complexity and space complexity. Evaluate the efficiency of breadth-first search and steepest-ascent hill-climbing. Conclude which technique is better to solve the Block World Problem mentioned in **Figure 1.1**.

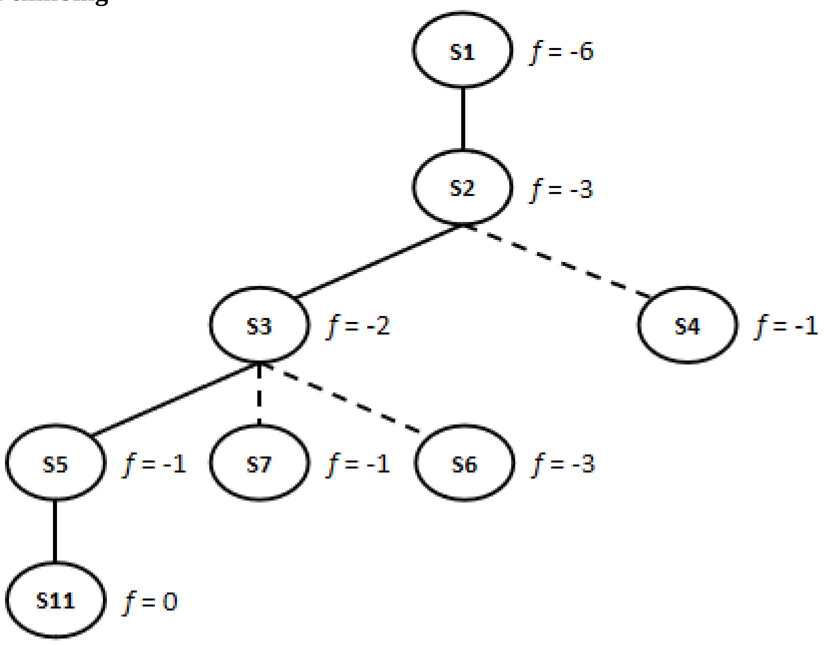
Answer:

a)
Step cost refers to the cost of an action. In other words, it is the cost of an action that transforms a state into another state. For instance, in the case of Block World problem, the step cost for all actions can be any but the same constant numeric value. However, in the case of path finding problem, step cost of an action might refer to the distance (km) or traveling time (seconds) between two states. Thus, most probably every step cost will have a different numeric value.
The value of step cost is 1.

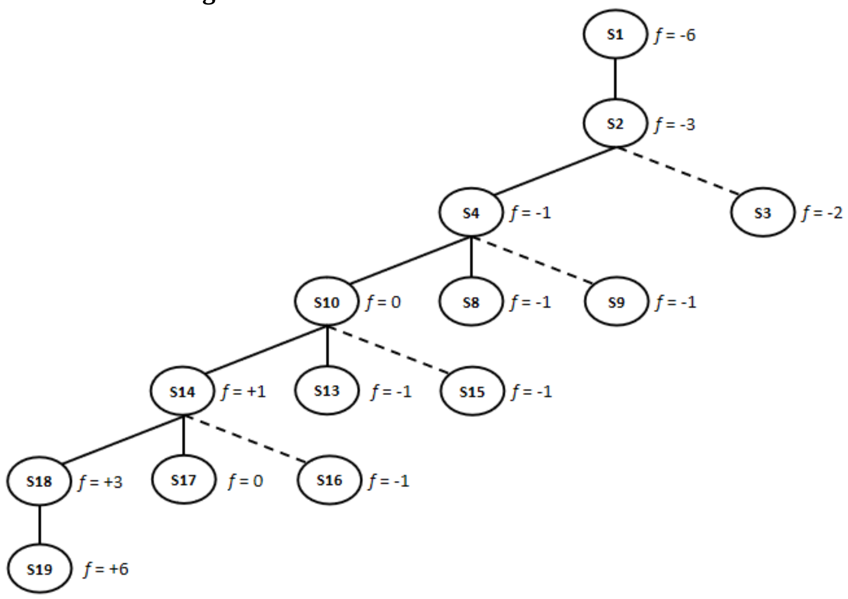
b)(i)
Local maximum refer to a state with no better child states or successors. The presence of local maxima makes a search fails to find a solution (e.g., 8-queen problem) or an optimal solution (e.g., travelling salesman problem), depending on the types of problem to be solved. For instance, in the case of 8-queen problem,
Foothill, plateau and ridge are 3 types of local maximum.

b)(ii)
The nature of its uphill moves that keep looking for a better successor but unable to backtrack to any least promising unexplored path increases its tendency to reach and terminate at a peak (or local maximum) where no neighbor has a higher value.

b)(iii)
Simple hill climbing



Steepest ascent hill climbing



c)

	Breadth-First Search	Steepest Ascent Hill Climbing
Completeness	Complete	Complete
Optimality	Optimal	Optimal
Time complexity	Has a higher time consumption because it explores all paths by expanding all successors.	Has a lower time consumption because it explores only the most promising path by expanding only the best successor.
Space complexity	Has a higher memory consumption because it keeps track of all expanded nodes	Has a lower memory consumption because it doesn't keep track of all expanded nodes

Steepest ascent hill climbing is a better approach because it has a better performance in terms of time and space efficiency.

2. **Figure 2** below shows an 8-puzzle problem, which requires rearrangement of the tiles to transform the order from start state to goal state. One is only permitted to slide the empty tile in **up**, **down**, **right or left**.

1	2	3
	4	5
7	8	6

Start → **Goal**

1	2	3
4	5	6
7	8	

Figure 2: The 8-Puzzle Problem

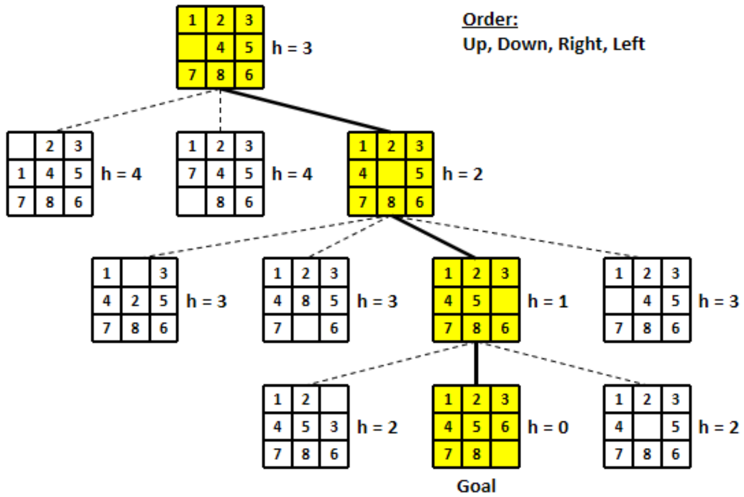
a) Suggest a heuristic function to produce a heuristic cost for a state. Demonstrate how such heuristic cost can be computed on the **start state**. Then perform best-first search.

b) Evaluate the efficiency of **breadth-first search** and **best-first search** in terms of completeness, optimality, time efficiency and space efficiency in solving the problem above.

c) Can hill-climbing find a solution for this problem? Draw a resulting search tree to support your answer.

Answer:

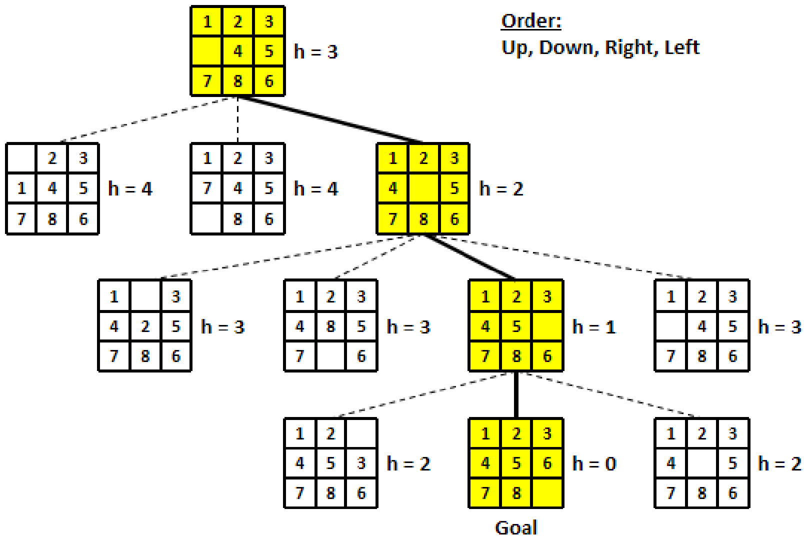
a)

Heuristic	Description
Hamming distance	The total number of misplaced tiles $f = 0 + 0 + 0 + 1 + 1 + 1 + 0 + 0 = 3$
Manhattan distance	The sum of the distances of each tile from its goal position $f = 0 + 0 + 0 + 1 + 1 + 1 + 0 + 0 = 3$
Best-First Search	

b)

	Breadth-First Search	Best-First Search
Completeness	Complete	Complete
Optimality	Optimal	Optimal
Time complexity	Has a higher time consumption because it explores all paths by expanding all successors.	Has a lower time consumption because it explores only the most promising path by expanding only the best successor.
Space complexity	Has a higher memory consumption because it keeps track of all expanded nodes	Has a lower memory consumption because it doesn't keep track of all expanded nodes

c)

Simple hill climbing	 Yes.

3. The following graph in Figure 1.1 shows all the nodes in a telecommunication network. The distance (in km) from one node to another is shown on the arc.

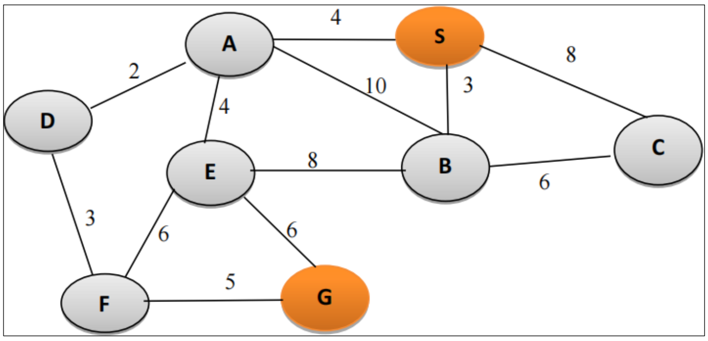


Figure 3: A search graph of a new LRT network

The Euclidean distance (in km), which is used as the heuristic cost (h) for different node, is provided in Table 1 below.

S	A	B	C	D	E	F	G
14	10	13	8	8	11	5	0

a) Assume that some data are to be sent from node S to node G using the shortest route. Describe the goal formulation and problem formulation.

b) Show the resulting search tree of A* search to find the shortest path from S to G. State the shortest path. (Remark: Ignore repeated nodes that have been visited previously)

c) Evaluate the efficiency of A* search in solving the path-finding problem above

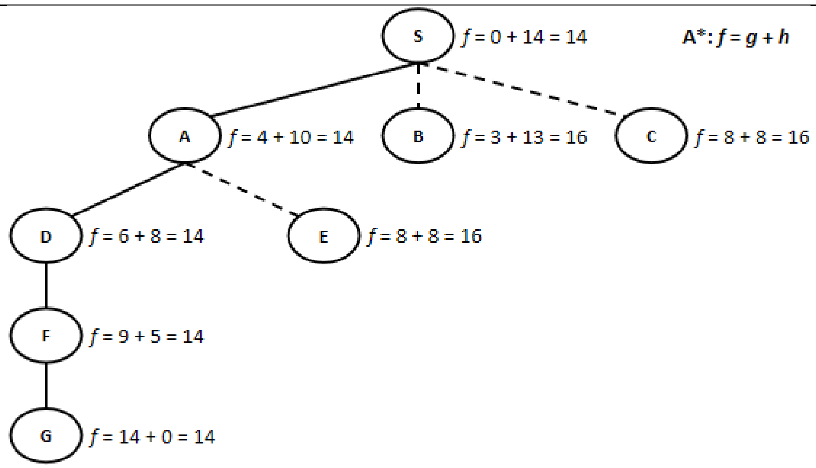
Answer:

a)

Goal	Node G
Optimal Solution	The shortest path used to send data from node S to node G.
Abstraction	Ignore the chance that some data might be lost during transmission.

Initial state	Node S
Successor function	The possible actions available to the agent that will change the state from the current state OR a function to generate a successor state (output) from a current state (input).e.g From A generates S etc. OR all possible movement available to the agent.
Goal test	A test to check if a node is node G.
Step cost	The cost of transforming a state to another state, indicated by the value of an edge connecting any two nodes. E.g., the step cost of transforming S to A is 4.
Path cost	The total step costs along a path. E.g., the path cost for S-A-D is 4 + 2 = 6.

b)



The shortest path discovered b A* is S-A-D-F-G

c)

	A*
Completeness	Complete
Optimality	Optimal
Time complexity	Time efficiency is high because it explores only the most promising path by expanding only the best successor.
Space complexity	Space efficiency is high because it keeps track only expanded nodes.

4. Consider 2 heuristic h_1 and h_2 of A* for the puzzle problem are defined as:
$h_1(n)$ = number of misplaced tiles
$h_2(n)$ = total Manhattan distance

3	1
2	

Start State

Goal State

1	2
3	

a) Illustrate the **state space** of the puzzle to reach the goal state based on:
- $h_1(n)$ 3
- $h_2(n)$ 4

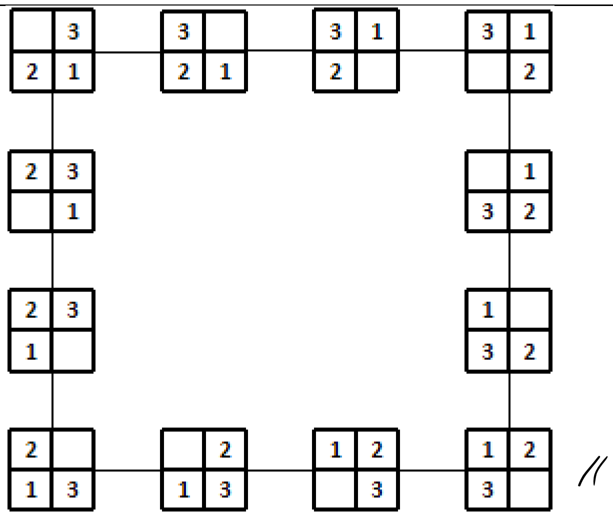
b) Show the resulting search trees of A* search to find the shortest path using the heuristic functions of:
- $h_1(n)$
- $h_2(n)$

You must clearly show the function cost, given that: $f(n) = h(n) + g(n)$, where $g(n)$ is the path cost.

Answer:

a)

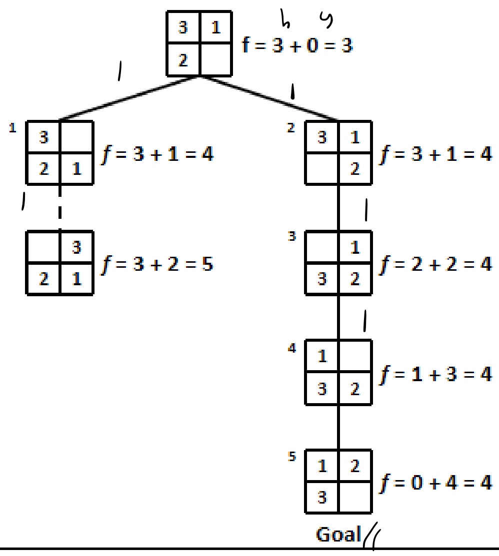
A **state space** is a set of all possible states that it can reach from the start state.



b)

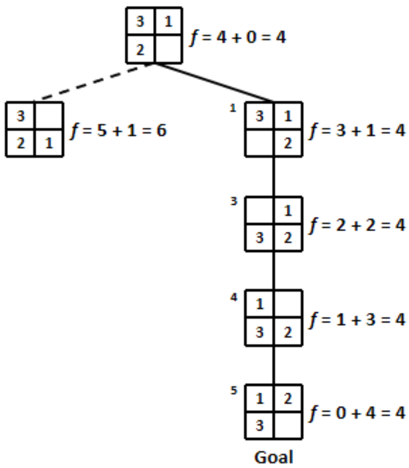
<table>
<tr><th>h₁(n)</th><th>h₂(n)</th></tr>
<tr><td>

$f(n) = h_1(n) + g(n)$



</td><td>

$f(n) = h_2(n) + g(n)$



</td></tr>
</table>