Tutorial 4 Informed Search

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 1.2 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.

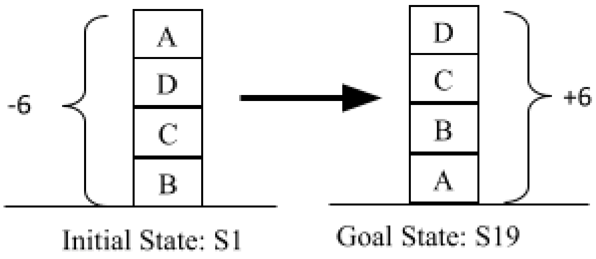


Figure 1.1

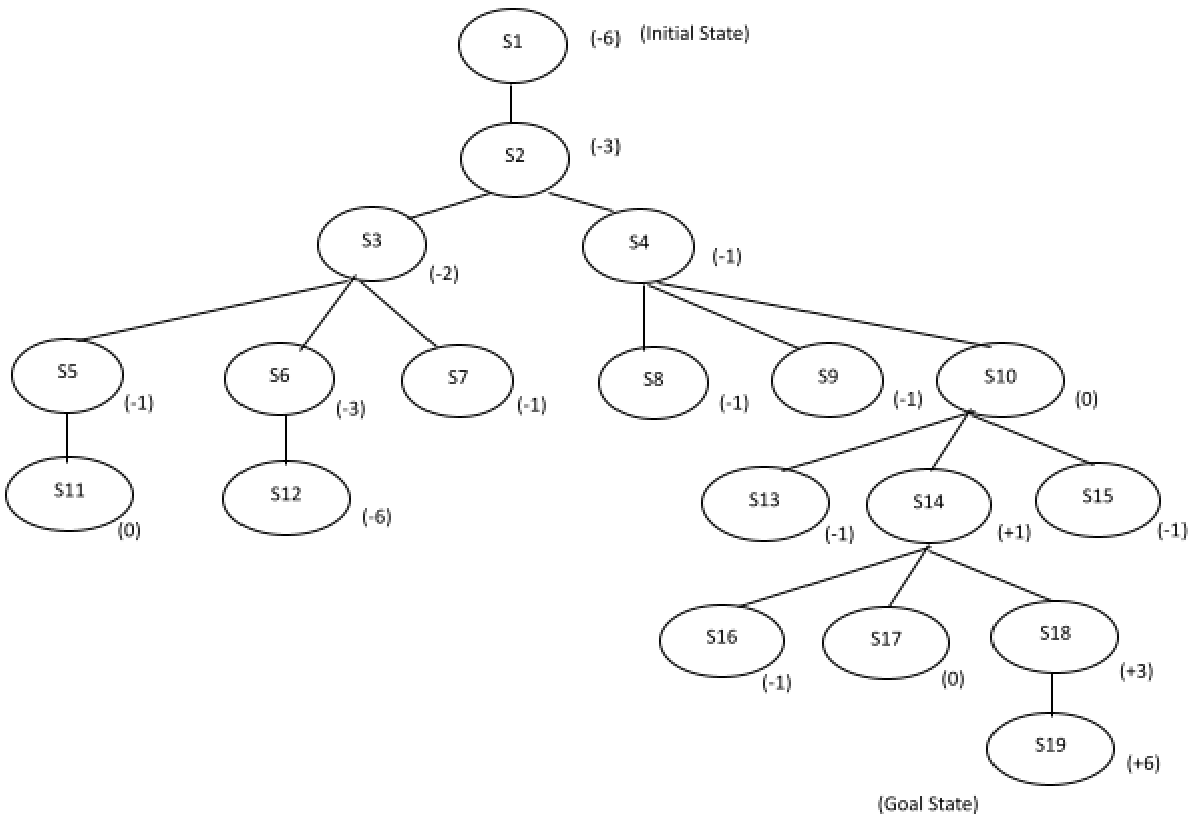


Figure 1.2

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

Answer:JOO HONG TEE

Step cost = 1

The robot will move the blocks one by one from initial state S1 to reach the goal state S19.Each time the robot moves the box, the step cost is one✔ and path cost will increase by one.The reach the goal, the path cost will be 6 to make all the blocks placed correctly with its supporting blocks.

(b) Hill climbing search is unable to guarantee completeness and optimality as it may be trapped into local maximum.

(i) Explain local maximum.

Answer:SWEE LOONG PHOO

- **Foothill -** a state that has no better move where a simple hill climber will never find a solution even though if one exists.✔
- **Plateau -** an area of state space where the heuristic values are equal where the hill climber will not know which direction to follow✔
- **Ridges -** an area of state space bordered by lower values in a descending search or higher values in ascending search where the ridge may slope very slowly towards the peak, making it a long route for the hill climber.✔

(ii) Discuss why hill climbing search always lead to a local maximum.

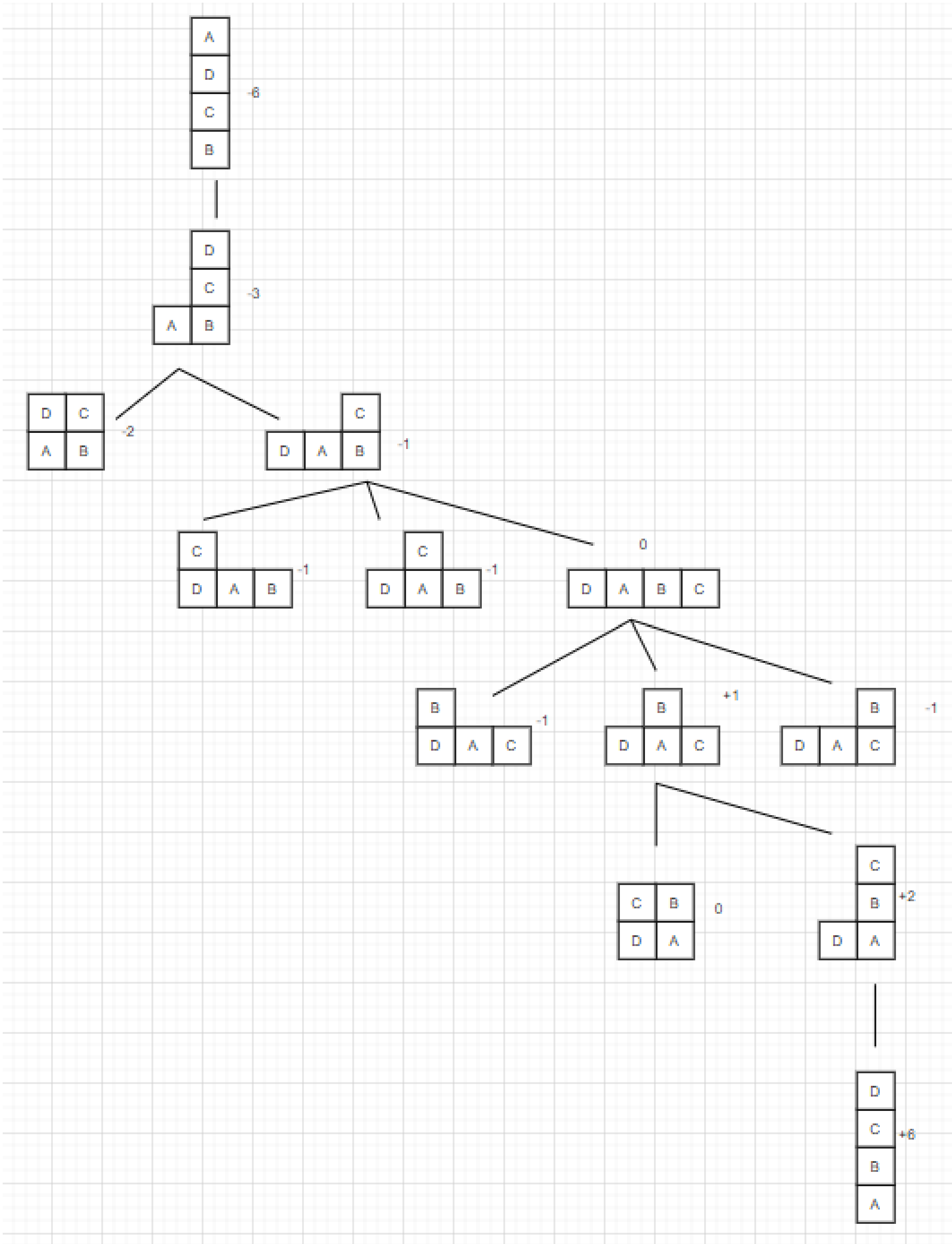
Answer:OH TZE KHAI JEFFREY

Hill climbing algorithm is a local search algorithm which continuously moves in the direction of increasing elevation/value to find the peak of the mountain or best solution to the problem. It will not choose a state with a lower elevation/value, and does not allow backtracking ✔as it does not remember the previous states. It also only considers neighbouring states and does not consider the global maximum. Hence, the search algorithm will always lead to a local maximum.

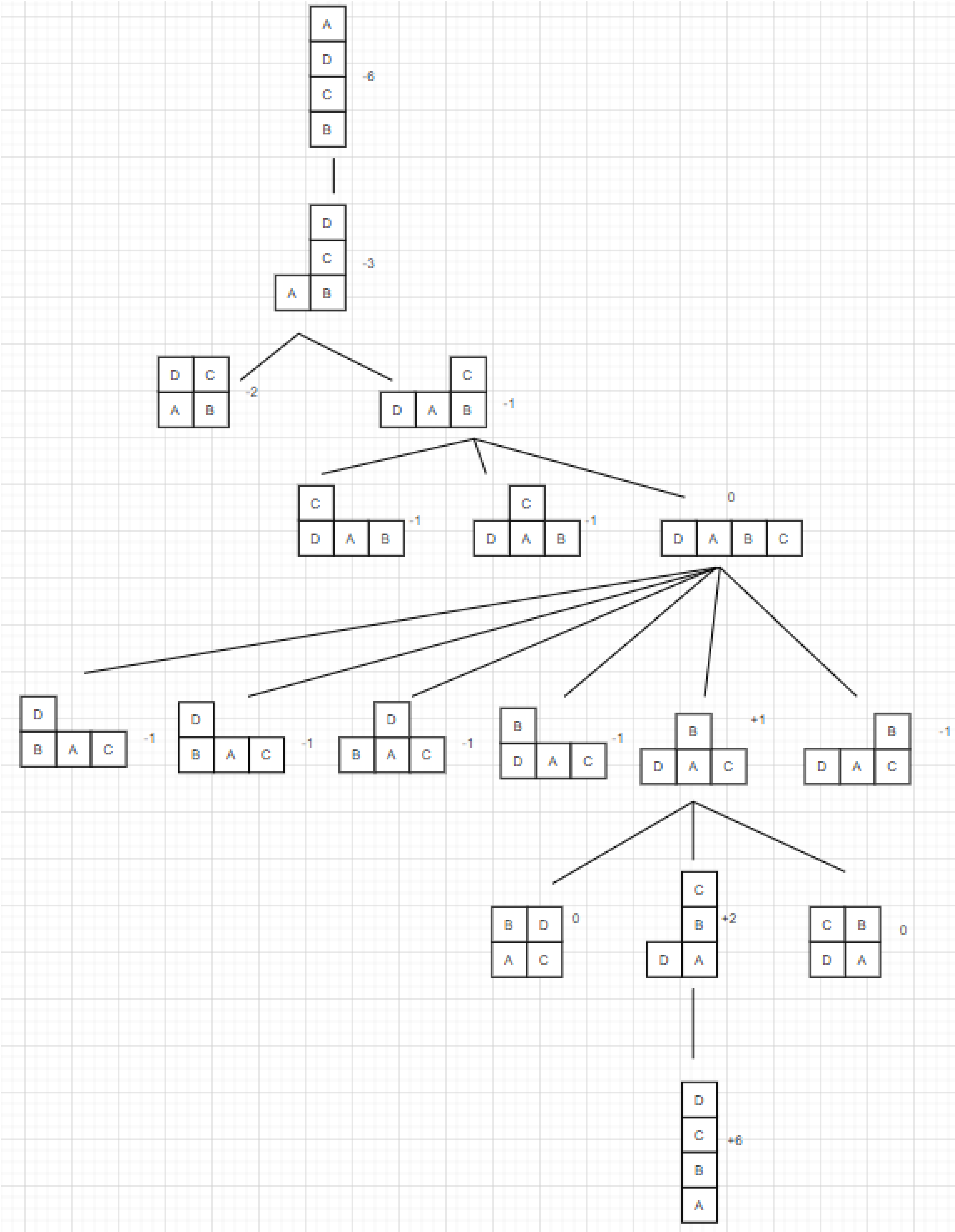
(iii) Use simple hill-climbing and steepest-ascent hill-climbing to search for the best path from S1 to S19 on the state space shown in Figure 1.2. Then for each search technique, draw the resulting search tree that shows the visited nodes. Show that hill-climbing technique can be trapped into a local maximum.

Answer:TZE KIN NG

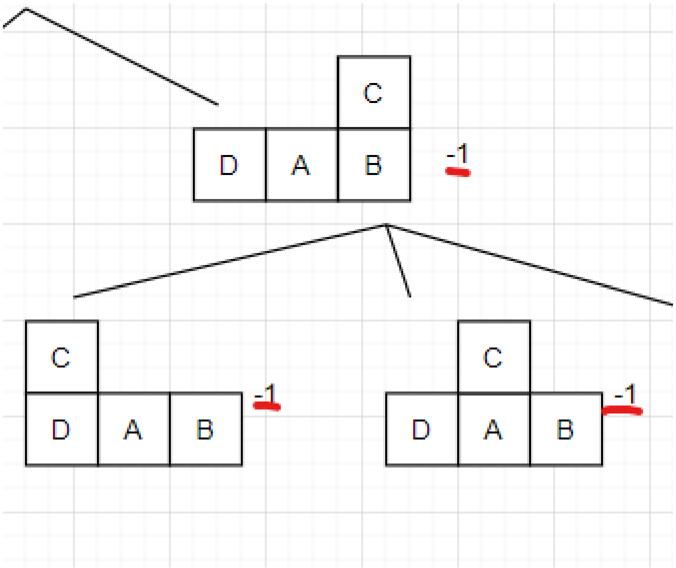
Simple HC:


A
D
C
B
-6
D
C
A B
-3
D C
A B
-2
C
D A B
-1
C
D A B
-1
C
D A B
-1
0
D A B C
B
D A C
-1
B
D A C
+1
B
D A C
-1
C B
D A
0
C
B
D A
+2
D
C
B
A
+6

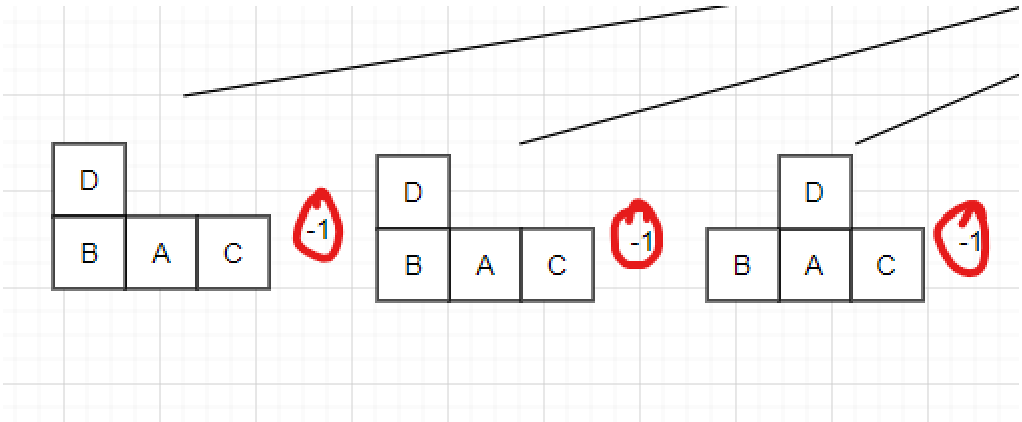
SAHC:



A
D
C
B
-6
D
C
A B
-3
D C
A B
-2
C
D A B
-1
C
D A B
-1
C
D A B
-1
0
D A B C
D
B A C
-1
D
B A C
-1
D
B A C
-1
B
D A C
-1
B
D A C
+1
B
D A C
-1
B D
A C
0
C
B
D A
+2
C B
D A
0
D
C
B
A
+6



SHC: Ridge occurs



SAHC: Plateau occurs.

(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:ZI YAN POH

	breadth-first search	steepest-ascent hill-climbing
completeness	Complete✔	Not complete, but in this case, solution is found✔
optimality	Optimal (shortest path)✔	Not optimal✔
time complexity	consume more time✔	lesser time compare to BFS✔ because you just expand the promising path
space complexity	Consume more memory O(bd)✔	Consume less memory O(1)-O(bd)✔

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 a tile **up, down, left or right** into the blank square.

1	**2**	**3**
	4	**5**
7	**8**	**6**

Start

□

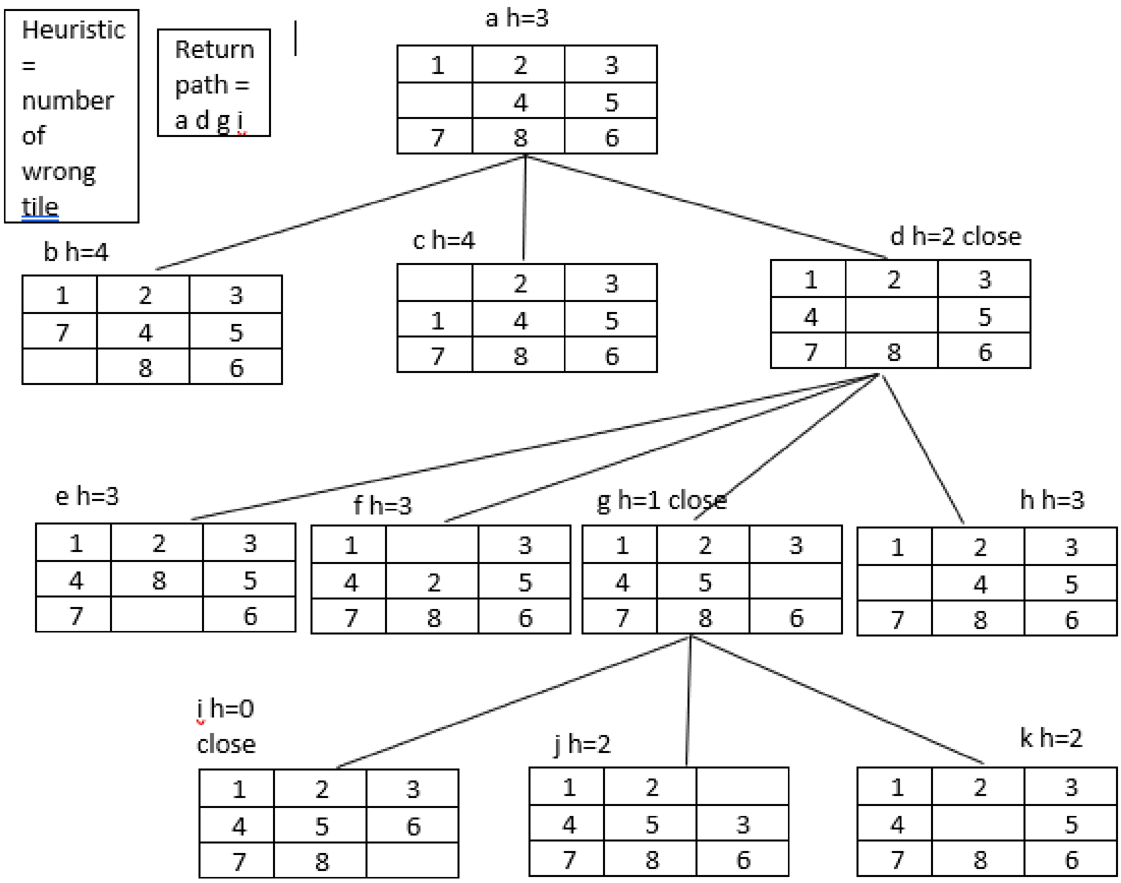
Goal

1	**2**	**3**
4	**5**	**6**
7	**8**	

Figure 2: The 8-puzzle problem

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.

Answer:YOON KHONG WONG



sequence= up, down , right, left✔

(i) 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.

Answer:CHIANG HANG CHAM

(ii) Explain why hill-climbing will fail in this problem.

Answer:SER KANG TAN

SHC will face the local maximum (foothill) as the H cost at Level 1 (4) is worse than the parent node (3)✔

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.

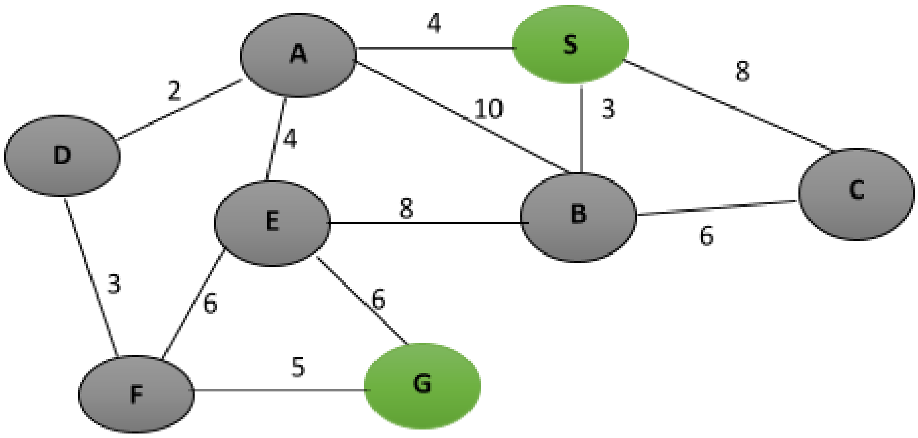


Figure 3 the search graph of a new telecommunication network

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

Table 1: The heuristic costs for different nodes

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

(i) 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.

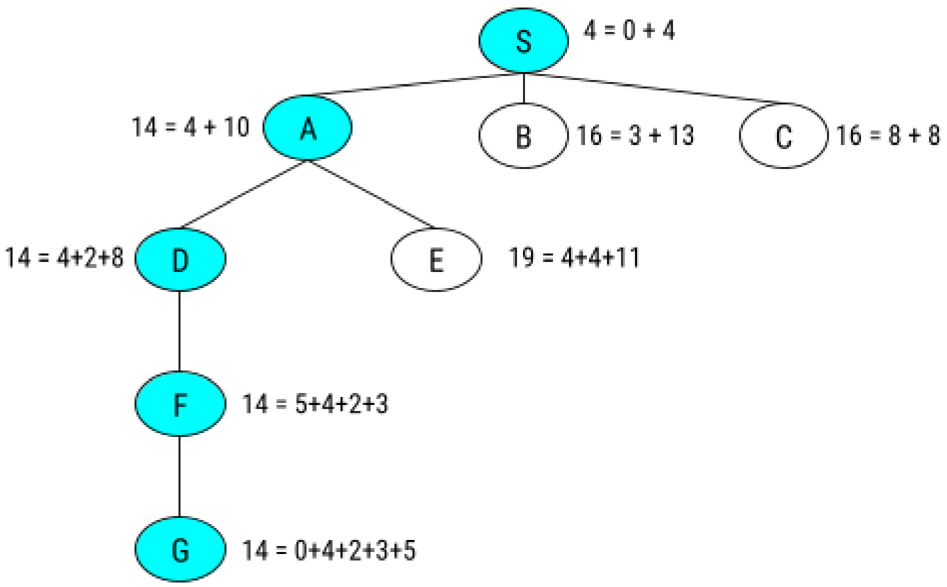
Answer:JIA SHIN LEE

Goal formulation	Problem formulation
- is the first step in solving problem	- is the process of deciding what actions and states to consider.
- goal :To reach node G✔	- initial state : start from node S✔
- optimal solution: reach the goal with minimal step/distance✔	- successor function: possible movement: e.g. node S to node B... based on the function cost = h(n) + g(n)✔

- abstraction: we don't care the time taken to reach the final destination.✔	- goal test :to check current state=goal state (node G)✔
	- path/step cost=distance between each of the nodes, path = total distance to reach the goal(node G)✔

(ii) **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)

Answer:XUE NIR YONG



Shortest path : S-A-D-F-G✔

(iii) Evaluate the efficiency of A* search in solving the path-finding problem above.

Answer:CHARUKESHI A/P CHANDRA SEKAR

Completeness : Complete because the branching factor is finite and cost at every action is fixed.✔
Optimality : The optimality of this A* search depends on how admissible the heuristic function is. Since the heuristic function is admissible so the A* tree search will find the least cost path and is optimal.is optimal ✔
Time complexity : The time complexity of A* search algorithm depends on heuristic function, and the number of nodes expanded is exponential to the depth of solution d. So the time complexity is O(b^d), where b is the branching factor. Therefore A* search is efficient in terms of time complexity because it consumes less time. ✔(consume less time)
Space complexity : The space complexity which is the maximum number of nodes that are stored in memory of A* search algorithm is O(b^d). This A* search algorithm is efficient in terms of space complexity as it consumes less memory.✔(less memory)

4. Consider 2 heuristic $h1$ and $h2$ of A* for the puzzle problem are defined as:

$h1(n)$= number of misplaced tiles

$h2(n)$= total Manhattan distance

3	1
2	

Start State

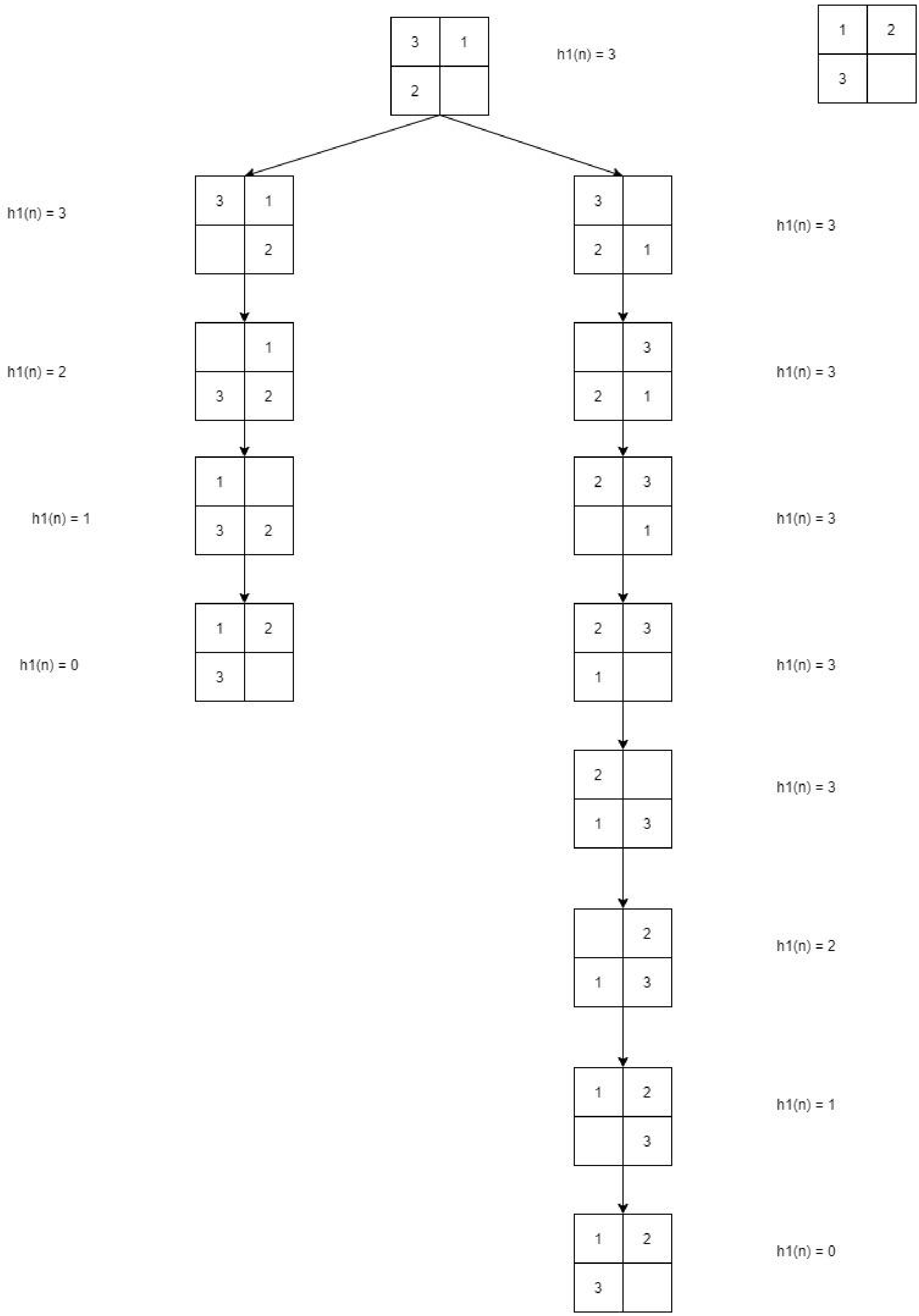
Goal State

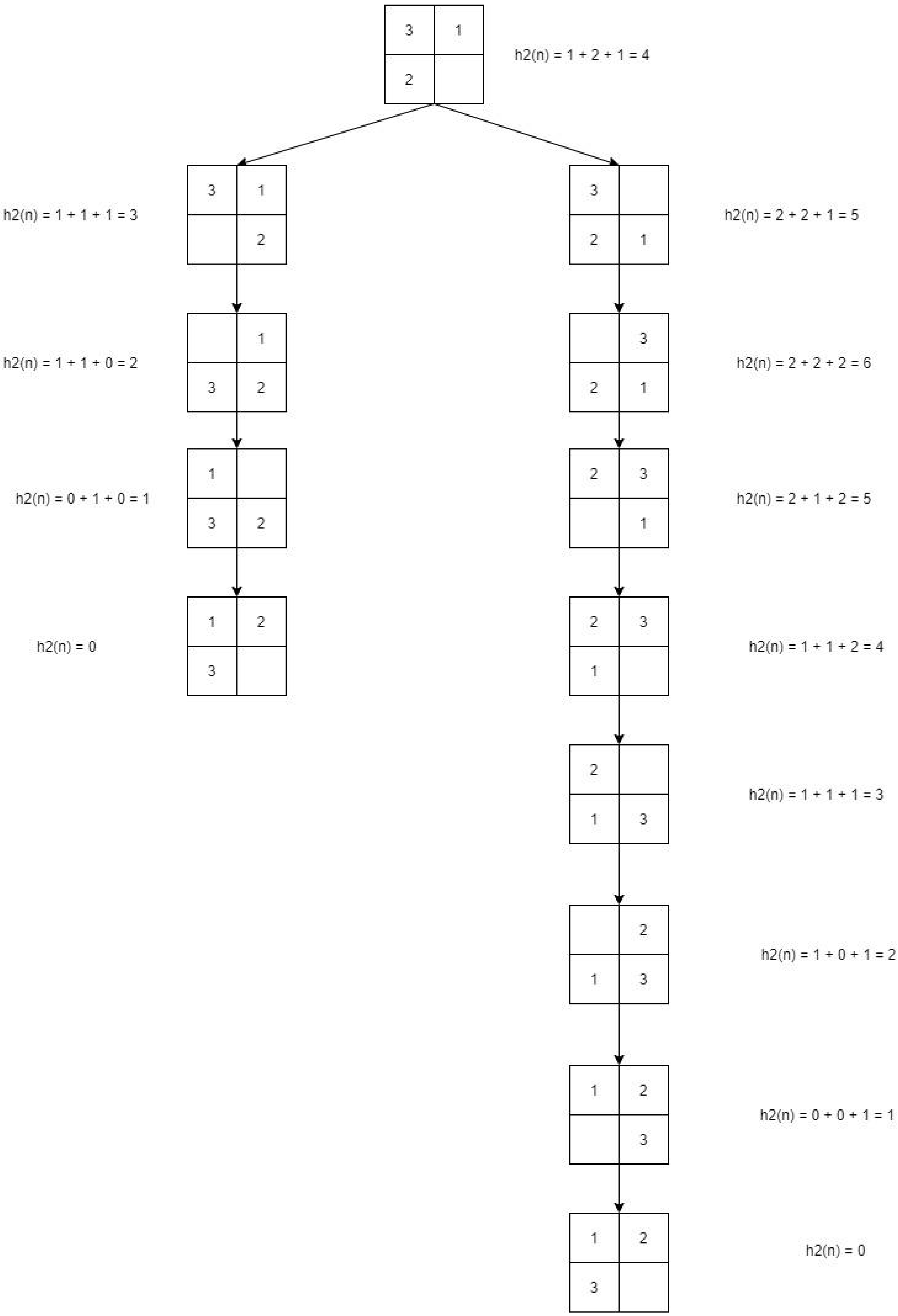
1	2
3	

(i) Illustrate the **state space** of the puzzle to reach the goal state based on:

- $h1(n)$
- $h2(n)$

Answer:WEI MENG NG


h1(n) = 3
h1(n) = 3
h1(n) = 3
h1(n) = 2
h1(n) = 3
h1(n) = 1
h1(n) = 3
h1(n) = 0
h1(n) = 3
h1(n) = 3
h1(n) = 2
h1(n) = 1
h1(n) = 0


3 1
2
h2(n) = 1 + 2 + 1 = 4
3 1
2
h2(n) = 1 + 1 + 1 = 3
3
2 1
h2(n) = 2 + 2 + 1 = 5
1
3 2
h2(n) = 1 + 1 + 0 = 2
3
2 1
h2(n) = 2 + 2 + 2 = 6
1
3 2
h2(n) = 0 + 1 + 0 = 1
2 3
1
h2(n) = 2 + 1 + 2 = 5
1 2
3
h2(n) = 0
2 3
1
h2(n) = 1 + 1 + 2 = 4
2
1 3
h2(n) = 1 + 1 + 1 = 3
2
1 3
h2(n) = 1 + 0 + 1 = 2
1 2
3
h2(n) = 0 + 0 + 1 = 1
1 2
3
h2(n) = 0

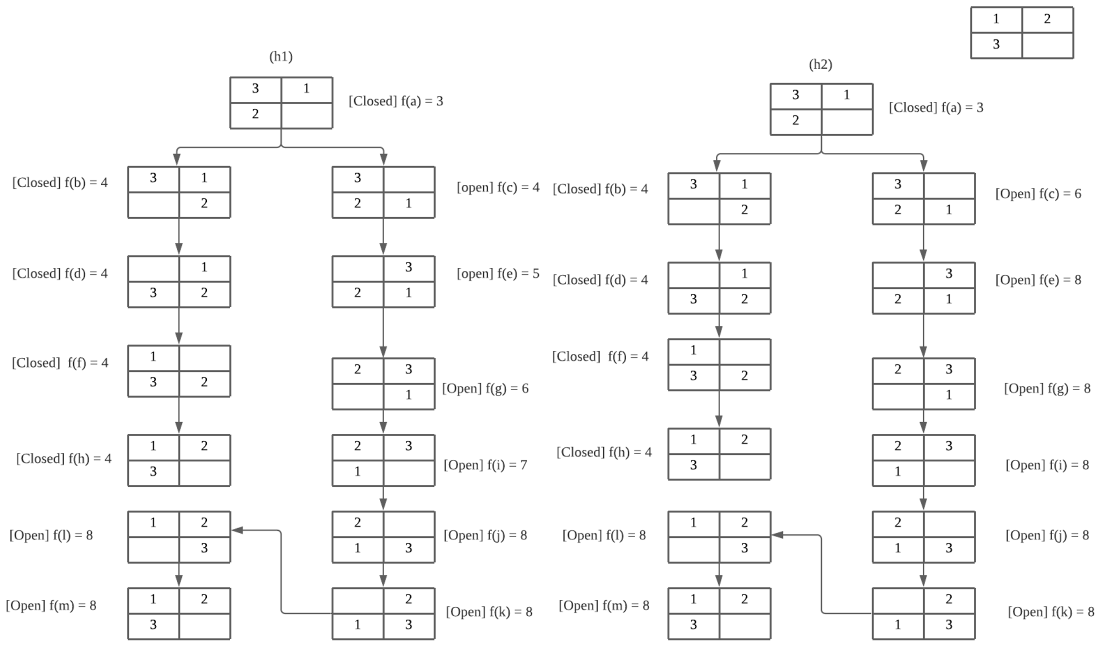
(ii) **Show the resulting search trees** of A* search to find the shortest path using the heuristic functions of:

- $h1(n)$
- $h2(n)$

You must clearly show the function cost, given that:

f(n) = h(n) + g(n), where *g(n)* is the path cost.

Answer:KIT YAO LOKE



(h1) = 4 (shortest path) add path cost into function cost

(h2) = 4