Page No.
Date

ASSIGNMENT No.:- 1(A)

NAME:-
Khan Zarrar Husain Zakir Husain

Roll No.:- 29

Class:- BE. IT

Batch:- I2

Sub.:- AI (ISLAB)

Q(1) Consider following definition of state space for some arbitrary problem. The number mentioned against the edges is cost to be incurred in moving from one node to the other in any direction. The number in red font mentioned against the node is heuristic function value.

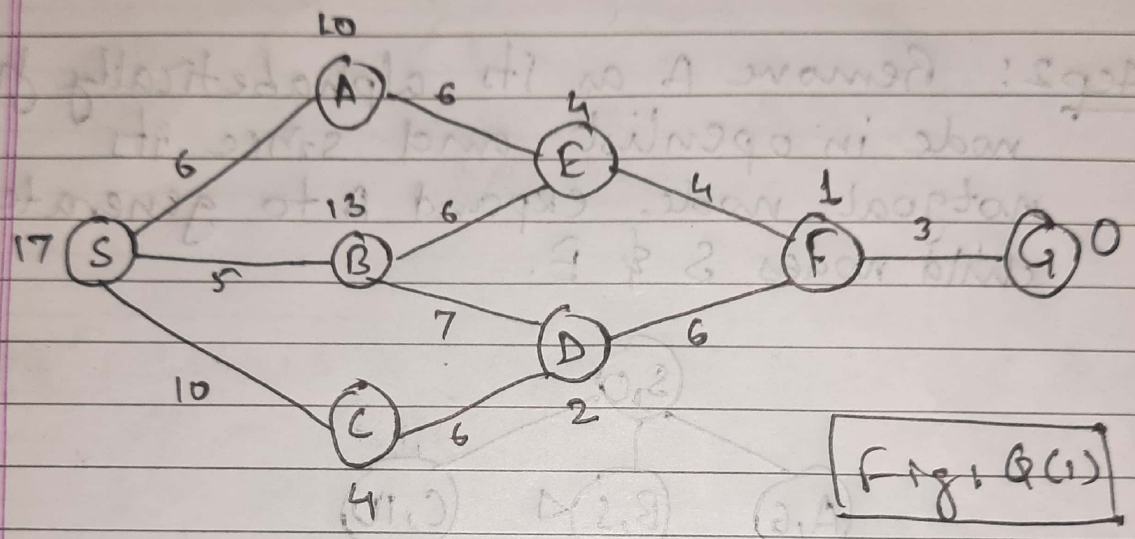


Fig: Q(1)

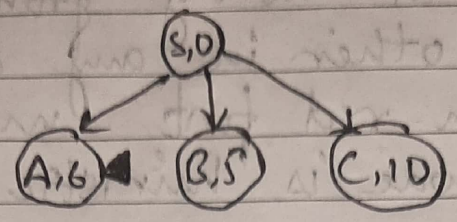
Using above data, solve following problems.

1 (1) Apply BFS on above graph. (Refer Fig: Q(1))

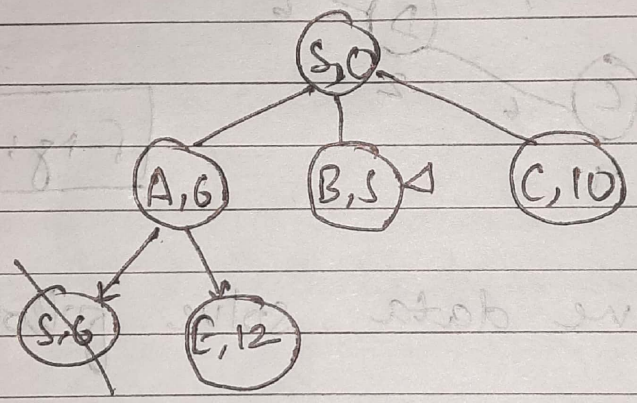
→ BFS (Breadth First Search) is a uniformed search algorithm.
We will show steps in Graph Search implementation of this algorithm ~~on~~ on above problems:

Step0: Put initial node S into openlist. Path cost for it is 0.

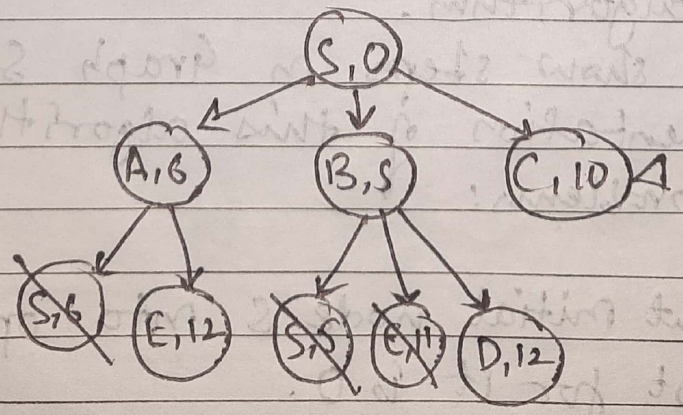
Step1: Remove S from openlist and since its not goal node (S,0) expand it to generate its child nodes A, B, C



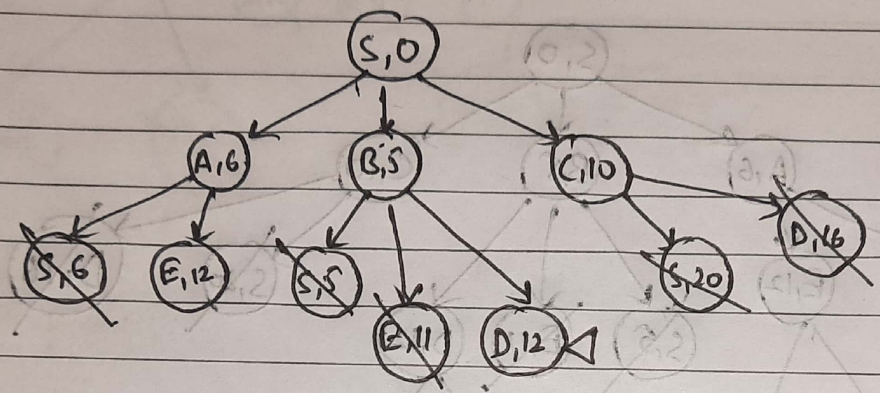
Step2: Remove A as its alphabetically first node in openlist and since its notgoal node expand A to generate child nodes S & E.



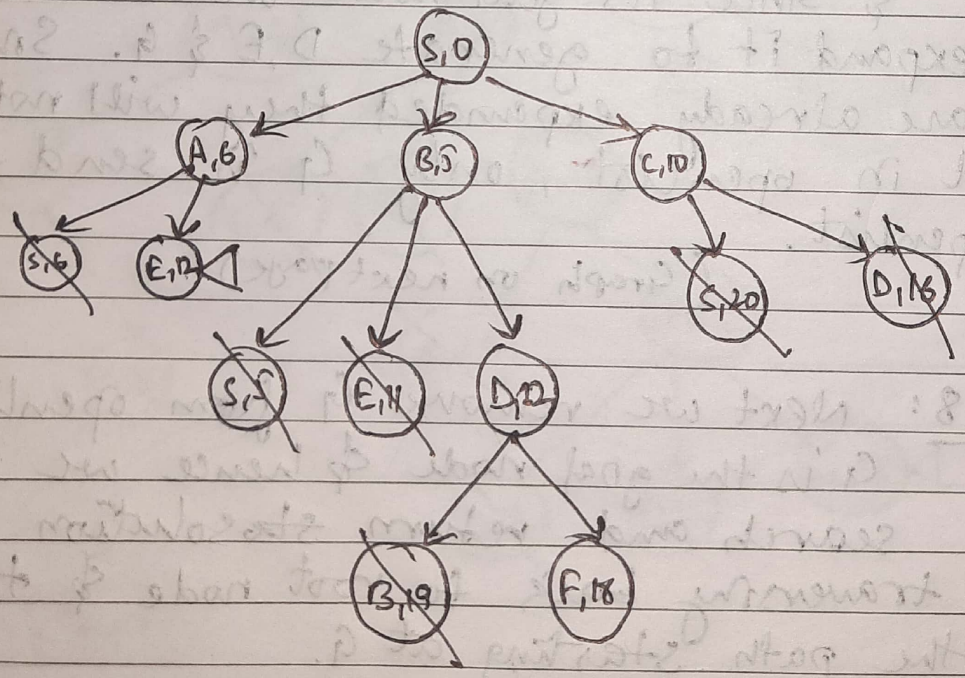
Step3: Remove B as it is alphabetically first node in openlist and since its notgoal node expand B to generate child nodes S, E & D.



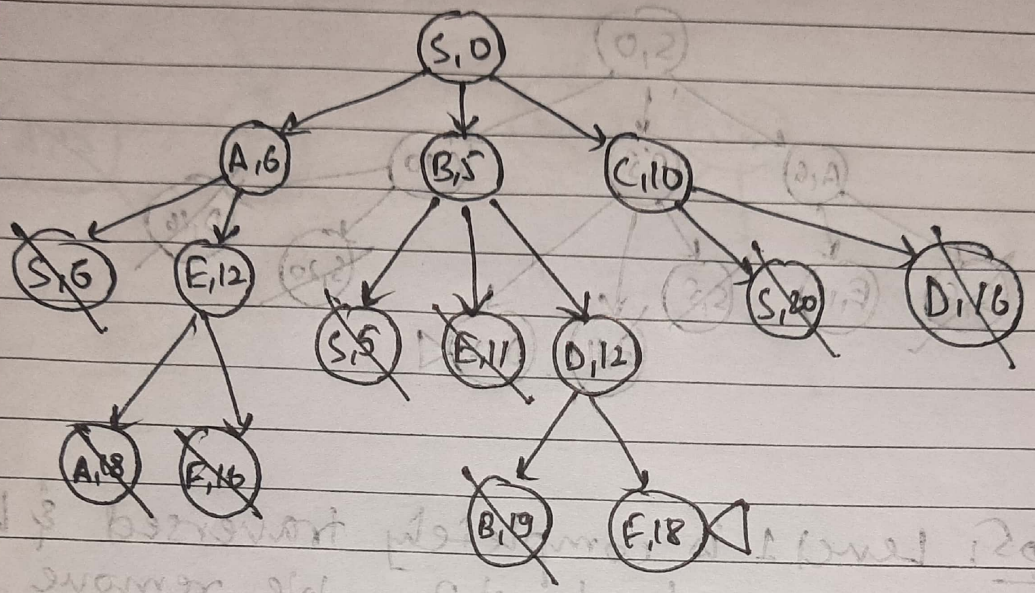
Step4: Remove C as it is alphabetically first node in openlist and since its not goal node, expand C to generate child nodes S & D. Since they are already in openlist don't put them in openlist.



Step5: Level 1 is completely traversed & hence we move to Level 2. We remove D from openlist and since its goal node, we will expand it to generate B & F.



Step 6:- Next we remove E from openlist & since its goal node we will expand it to generate A & F. Since they are already expanded they will not appear/ ~~to~~ be put in openlist.

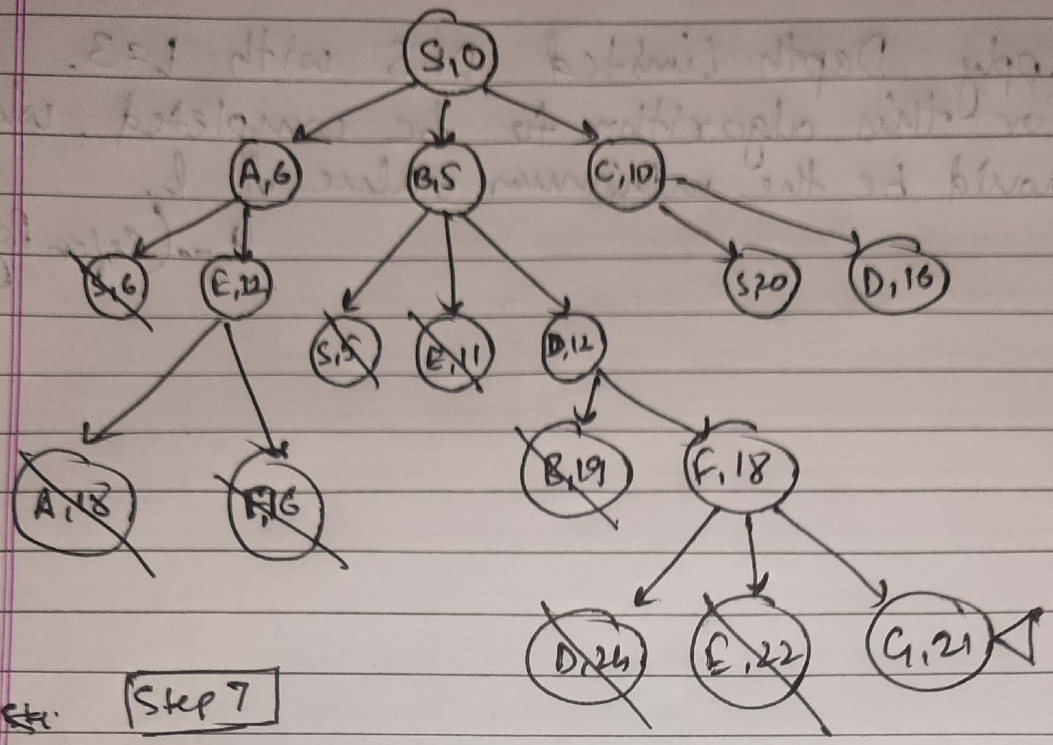


Step 7:- Next we remove F from openlist & since its goal node we will expand it to generate D, E & G. Since D & E are already expanded they will not be put in openlist, only G is send to openlist.

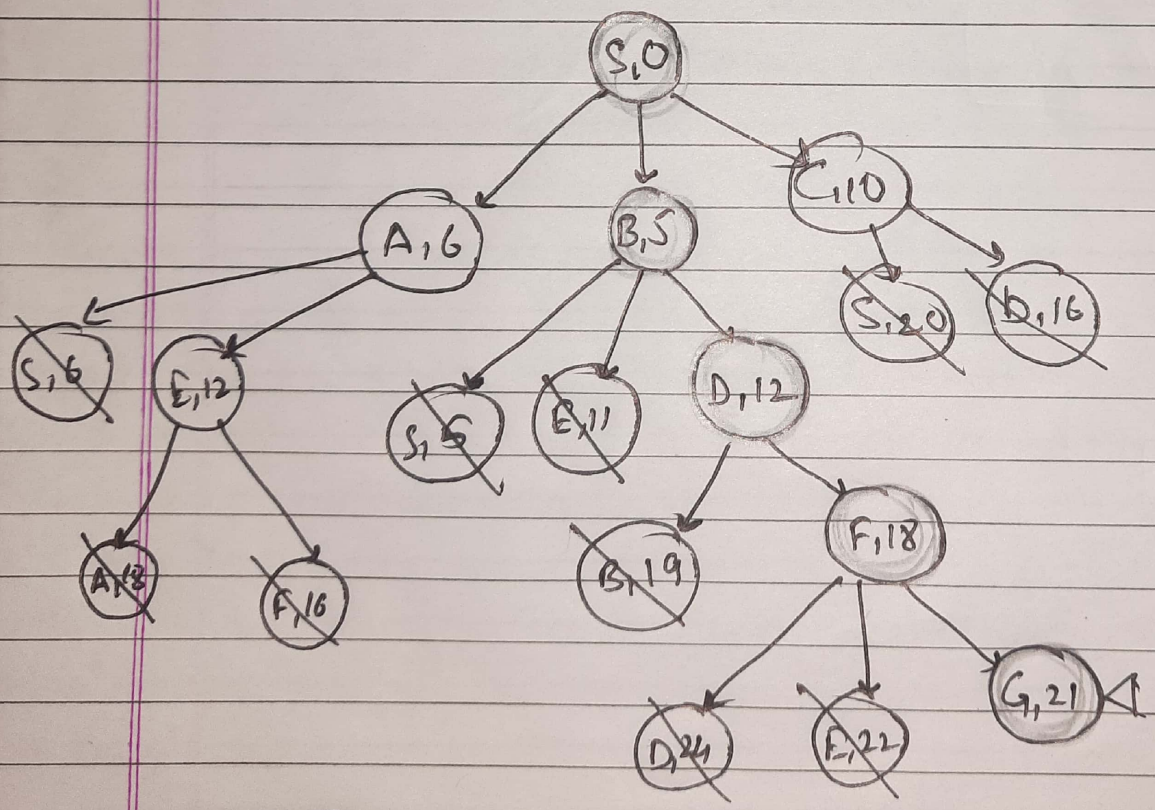
(Graph on next page)

Step 8: Next we remove G from openlist. G is the goal node & hence we stop search and return ~~sto~~ solution by traversing back to root node & traversing the path starting at G.

(Graph on next page)


S,0
A,6
B,5
C,10
S,6
E,12
S,5
E,11
D,12
S,20
D,16
A,18
F,16
B,19
F,18
D,24
E,22
G,21

Step 7


S,0
A,6
B,5
C,10
S,6
E,12
S,5
E,11
D,12
S,20
D,16
A,18
F,16
B,19
F,18
D,24
E,22
G,21

Step 8

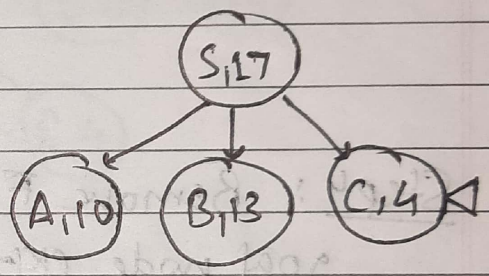
1 (4) Apply Best First Search & clearly show all steps using search tree. ...(Refer Fig: Q(1))

→ Initialization: Compute f-score for S & put it in the openlist.

f-score S: $f(S) = h(S) = 17$

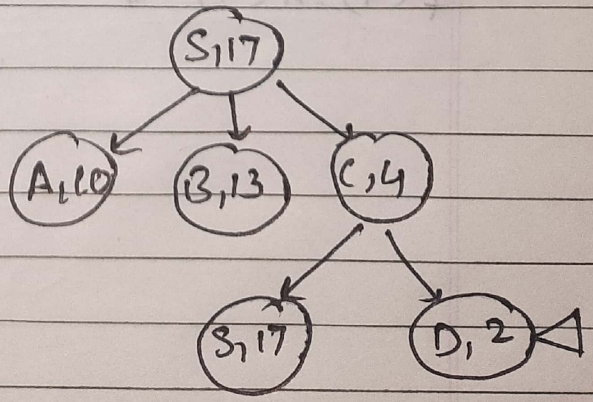
Step1: Remove S from openlist & since its not goal node, expand it. For each successor A, B, C compute f-score and put them into openlist with increasing order of f-score.

f-score of successors
$f(A) = h(A) = 10$
$f(B) = h(B) = 13$
$f(C) = h(C) = 4$



Step2: Remove C from open list & since its not goal node, expand it. For each successor S, D ~~C,B,A~~ compute f-score and put them in openlist with increasing order of f-score.

F-score of Successors
$f(S) = h(S) = 17$
$f(D) = h(D) = 2$



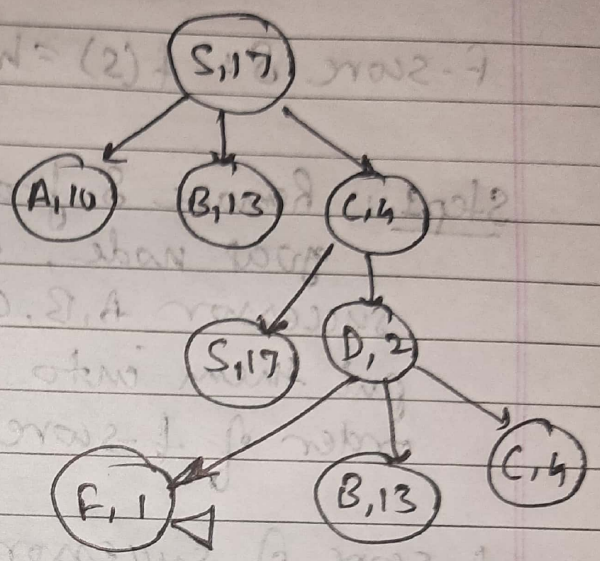
Step 3: Remove D from openlist and since its not goal node expand it. For each successor C, B, F compute f-score & put them in openlist with increasing order of f-score.

F-score of successors:

$f(C) = h(C) = 4$

$f(B) = h(B) = 13$

$f(F) = h(F) = 1$



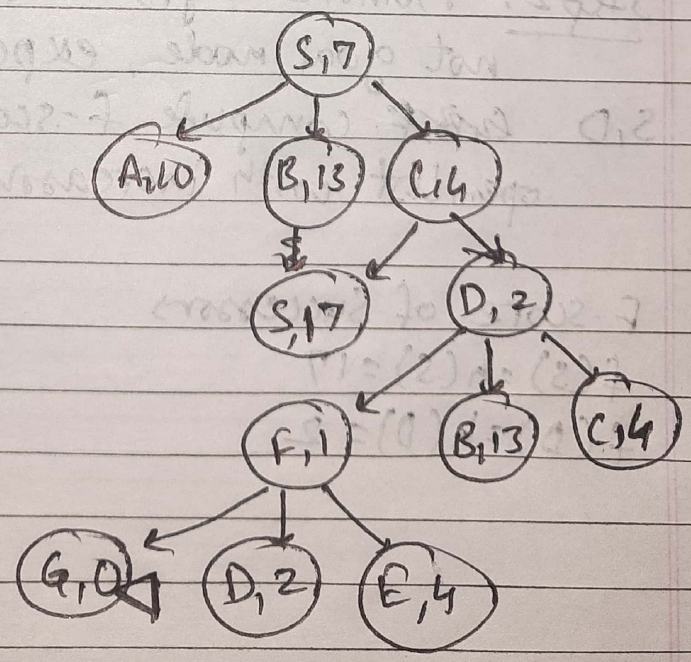
Step 4: Remove F from openlist & since its not goal node expand it. For each successor D, E, G compute f-score & put them in openlist with increasing order of f-score.

f-sore of successors:

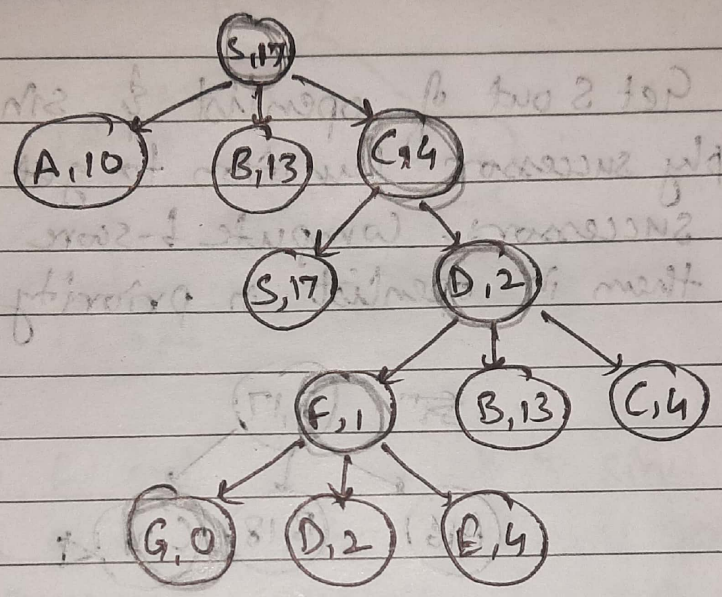
$f(D) = h(D) = 2$

$f(E) = h(E) = 4$

$f(G) = h(G) = 0$



Step5: Remove G from openlist and since its goal node it passes goal test. Stop search and traverse backward following parent pointers to the root node search tree, reverse the path traverse return it as solution to problem.



Solution is $S \rightarrow C \rightarrow D \rightarrow F \rightarrow G$

with

Solution cost $= 10 + 6 + 6 + 3 = 25$

This solution is not an optimal solution.

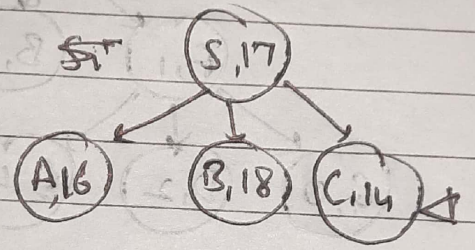
1 (5) Apply A* algorithm & clearly show all the steps using search tree. ...(Refer fig: Q(1))

→

Step1: Calculate f-score for S(f(S) = 0 + 17 = 17) put S in open list.

(S, 17)

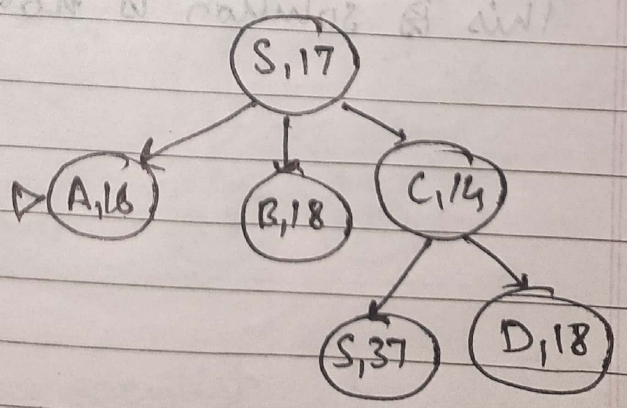
Step2: Get S out of openlist & since its not goal apply successor function to get A, B & C as its successors. Compute f-score of them & put them in openlist in priority order.



f(A) = ~~h(A)~~ = 6 + 10 = 16
f(B) = 5 + 13 = 18
f(C) = 10 + 4 = 14

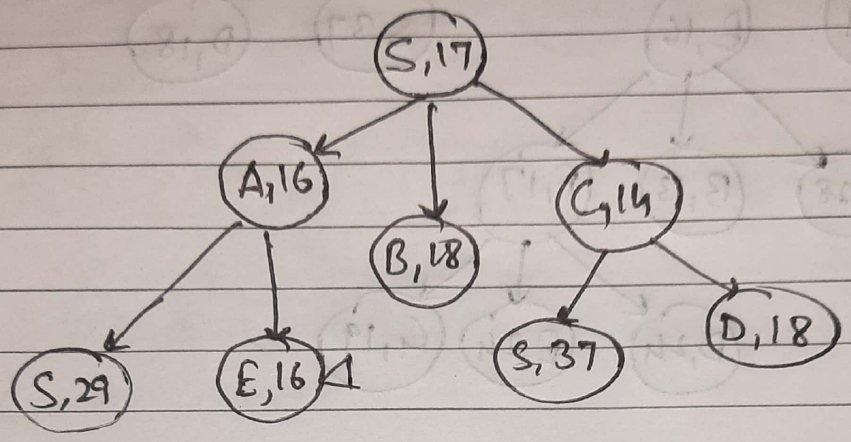
Step3: Get C out of openlist & since its not goal node, apply successor function to get S, D as its successors. Compute f-score for them & put them in openlist in priority order.

f(S) = (6+6) + 17 = 29
f(D) = (6+6) + 4 = 16
f(D) = = 18



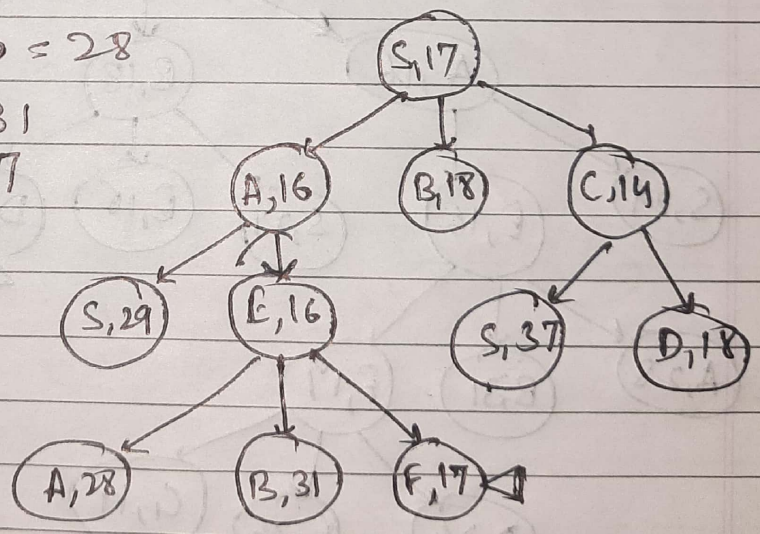
Step 4: Get A out of openlist & since its not goal, apply successor function to get S & E, as its successors. Compute f-score for them and put them in openlist in priority order.

$f(E) = (6+6)+4 = 16$



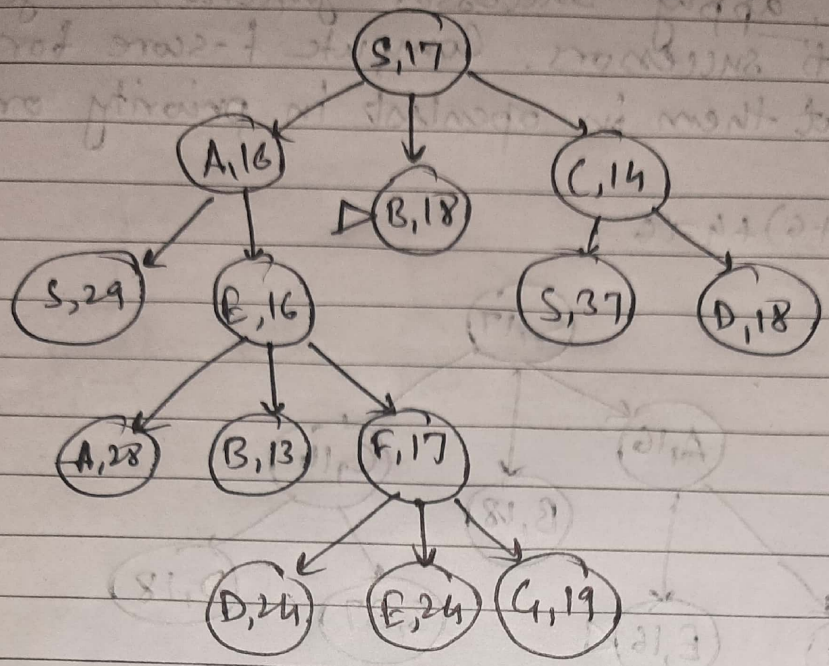
Step 5: Get E out of openlist & since its not goal apply successor function to get A, B & F as its successors. Compute f-score for them & put them in openlist in priority order.

$f(A) = (6+6+6)+10 = 28$
$f(B) = (6+6+6)+13 = 31$
$f(F) = (6+6+4)+1 = 17$

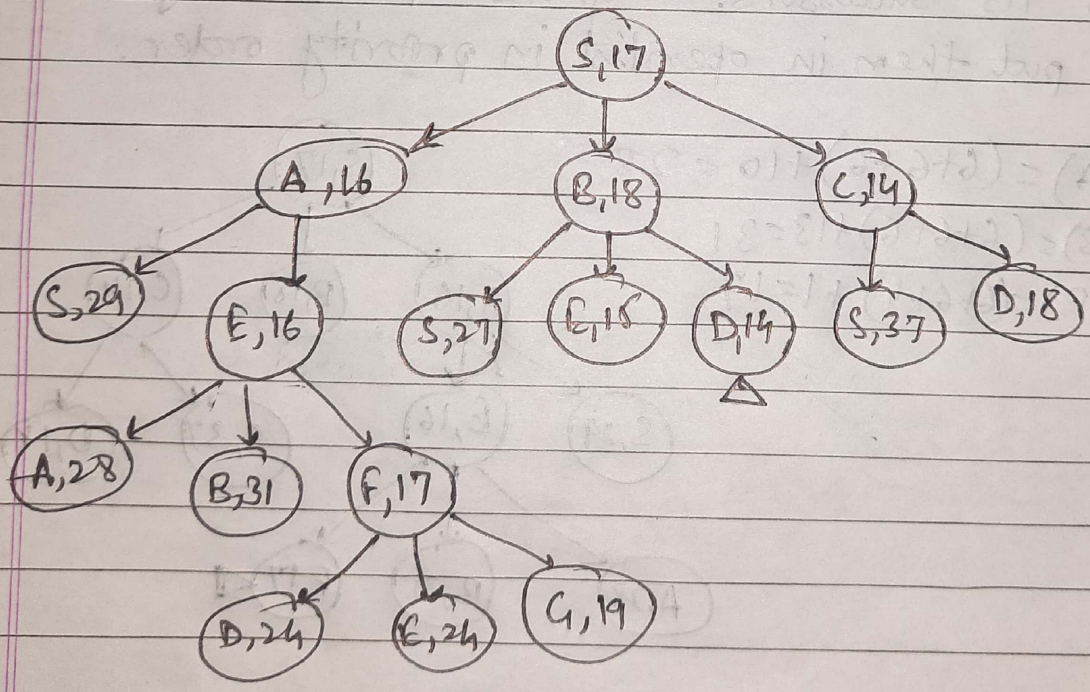


Step 6: Get F out of openlist & since its not goal, apply successor function to get D, E & G as its successors. Compute ~~f source~~ f-score for them & put

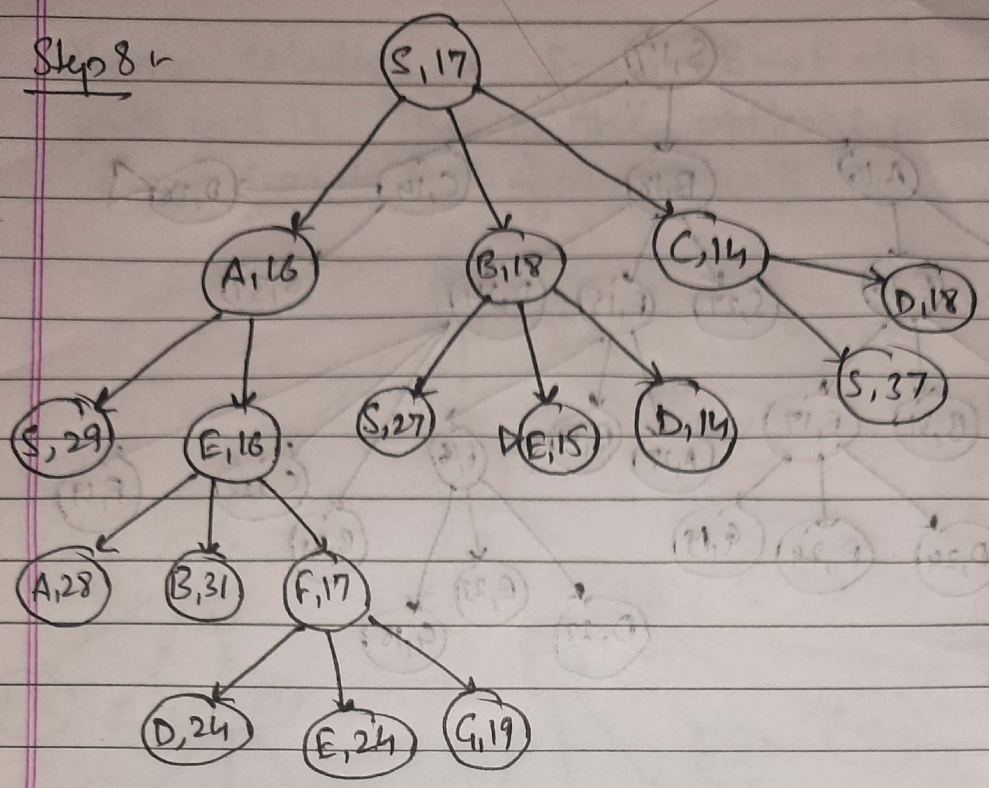
them in openlist in priority order.



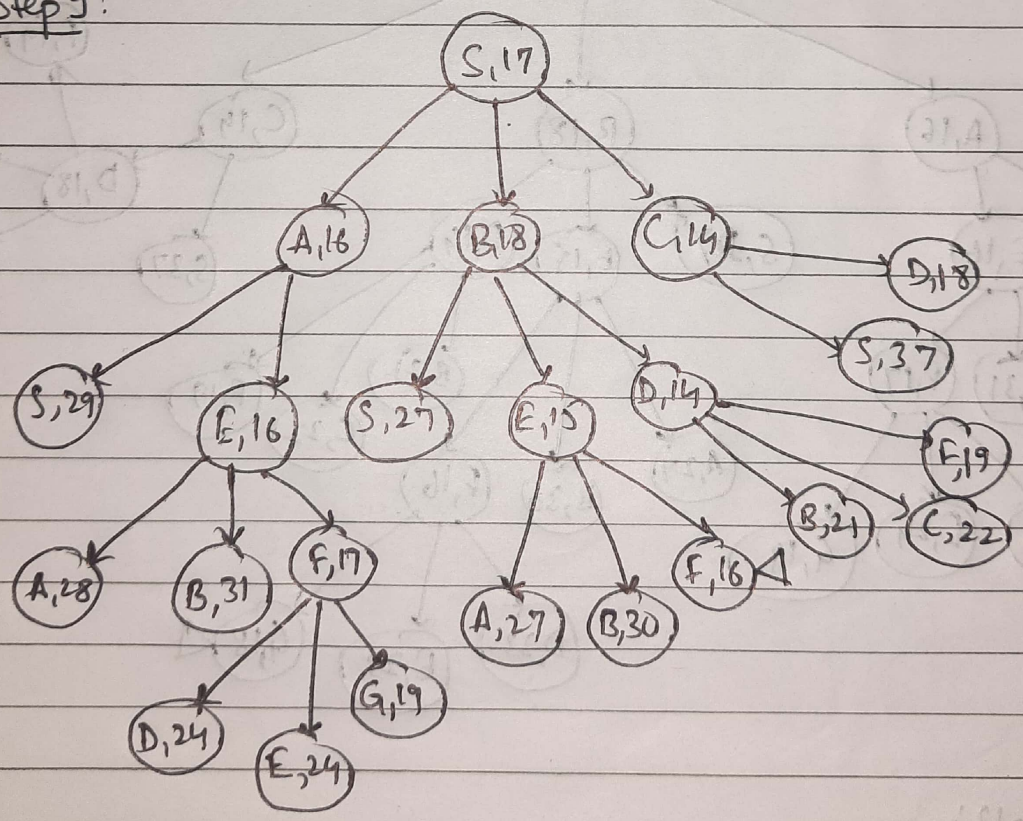
Step 7: Get B out of open list & put S, E, D in openlist in priority order.



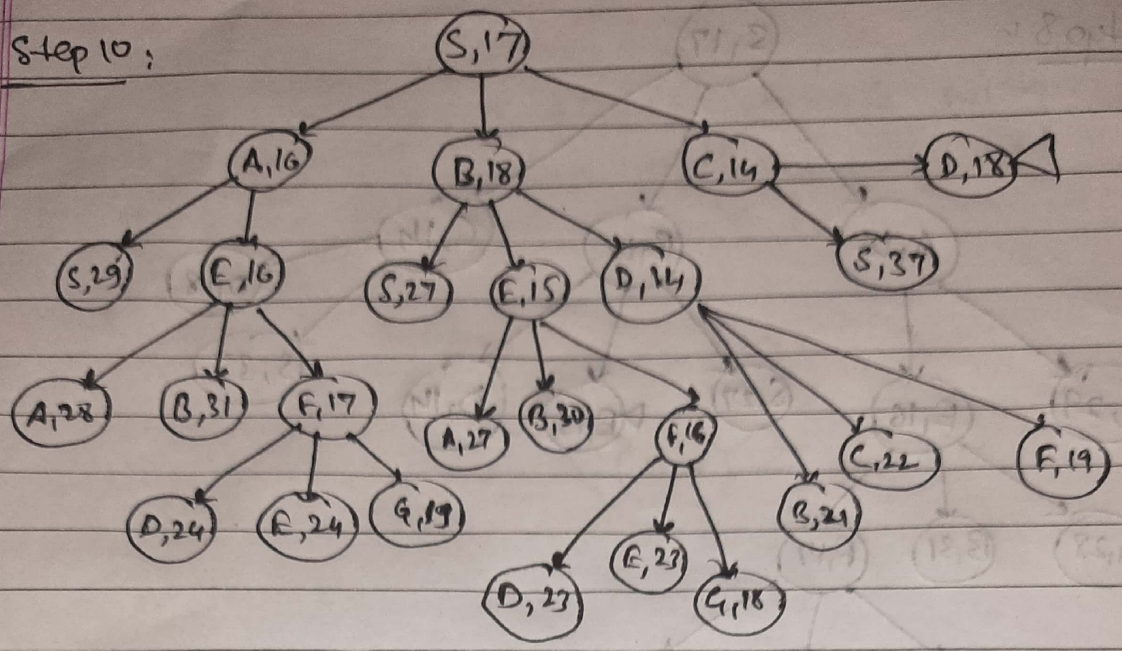
Step 8 :-


S,17
A,16
B,18
C,14
D,18
S,37
S,29
E,16
S,27
E,15
D,14
A,28
B,31
F,17
D,24
E,24
G,19

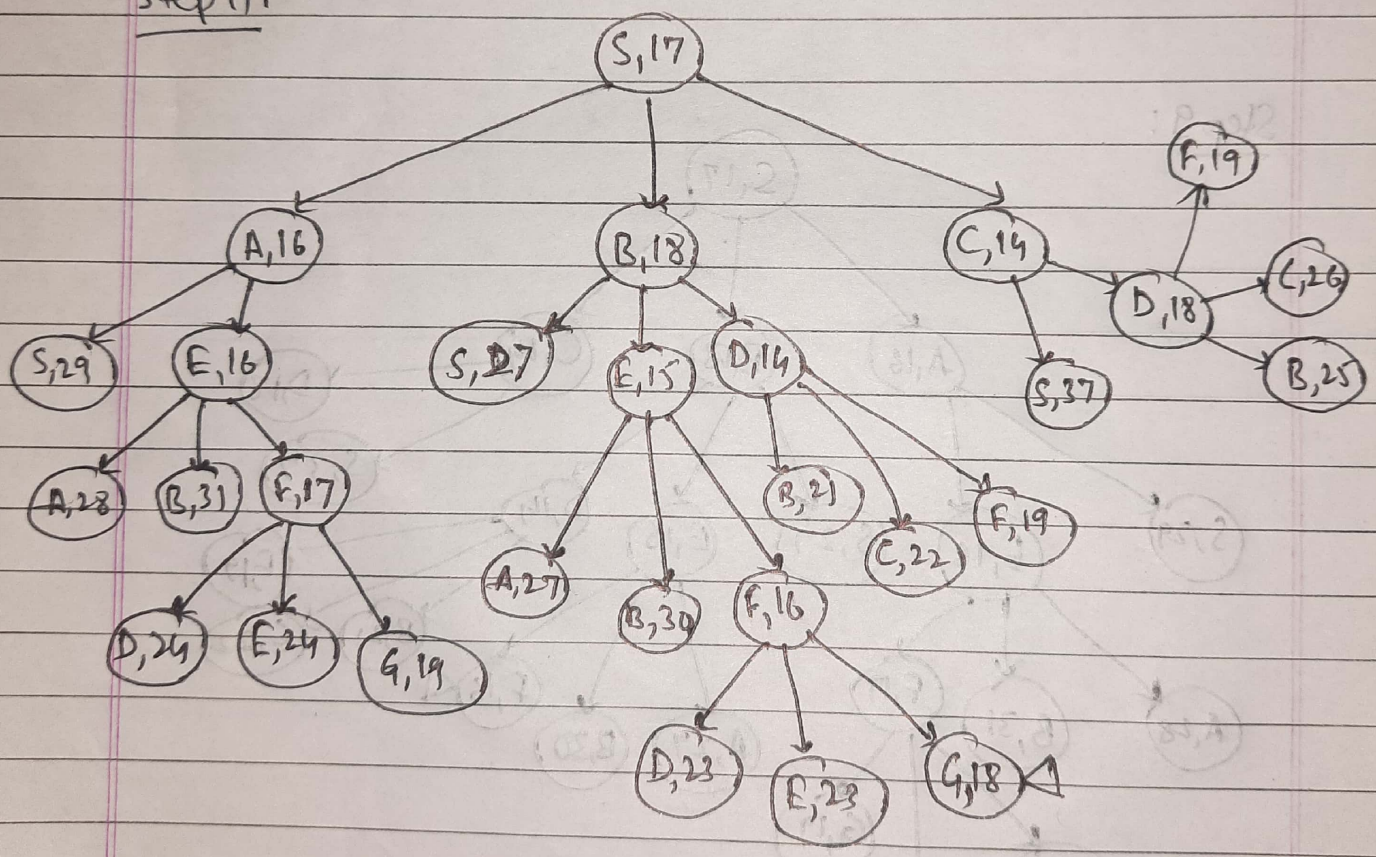
Step 9 :


S,17
A,16
B,18
C,14
D,18
S,37
S,29
E,16
S,27
E,15
D,14
F,19
B,21
C,22
A,28
B,31
F,17
F,16
A,27
B,30
G,19
D,24
E,24

Step 10:



Step 11:

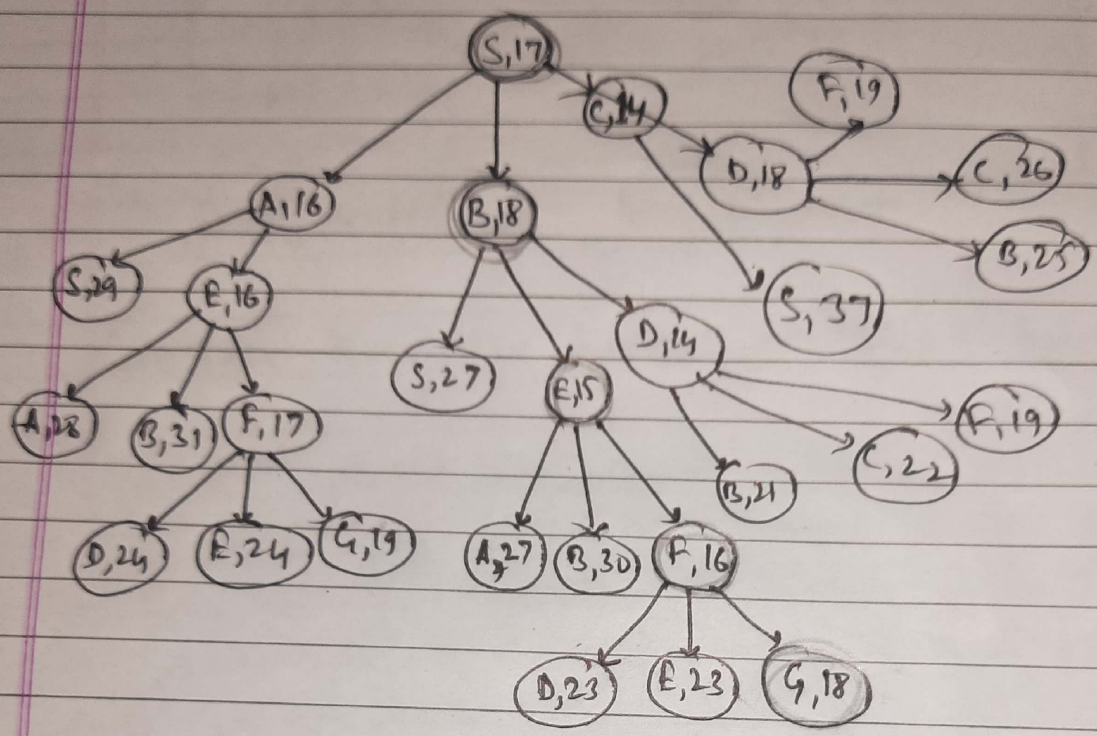


Step 12:

Remove G,18 from open list, since G is goal node goal test results in true and we will find the solution and return it. We traverse backwards towards the root from G,18 and

Hence solution is $S \rightarrow B \rightarrow E \rightarrow F \rightarrow G$ with path cost 18. This is the optimal path cost

Final Search Tree after A* will be:



Page No.
Date

Q(2) Consider the following instance of 8 puzzle problem:

8	7	6
2	1	5
3	4	-

Initial configuration

-	8	7
2	1	6
3	4	5

Goal configuration

Fig: Q(2)

Consider Hueristic functions defined below:

h1: Misplaced tiles count except space
h2: Correctly placed tiles count except space
h3: Sum of Manhattan distance between current and correct position of all times except space.

Answer following questions:

(a) In 8 puzzle problem we are concerned with getting to goal configuration within least number of steps. All moves are thus equally costly. Define g(n) in your own words. What will be the cost of 6 step solution to some arbitary 8 puzzle instance? ...(Refer Fig: Q(2))

→ The lowest path cost g(n) can be the cost to reach the goal configuration in least steps. In our case, we can reach the final ~~cof~~ configuration in at least 4 moves: UP, UP, LEFT, LEFT.

Since all the moves are equally costly, we compute $g(n)$ as

$$g(n) = 1+1+1+1$$
$$g(n) = 4$$

Consider the following arbitary 8 puzzle instance which gives solution in 6 steps:

8	7	6
2	1	5
-	3	4

The solution can be represented as:

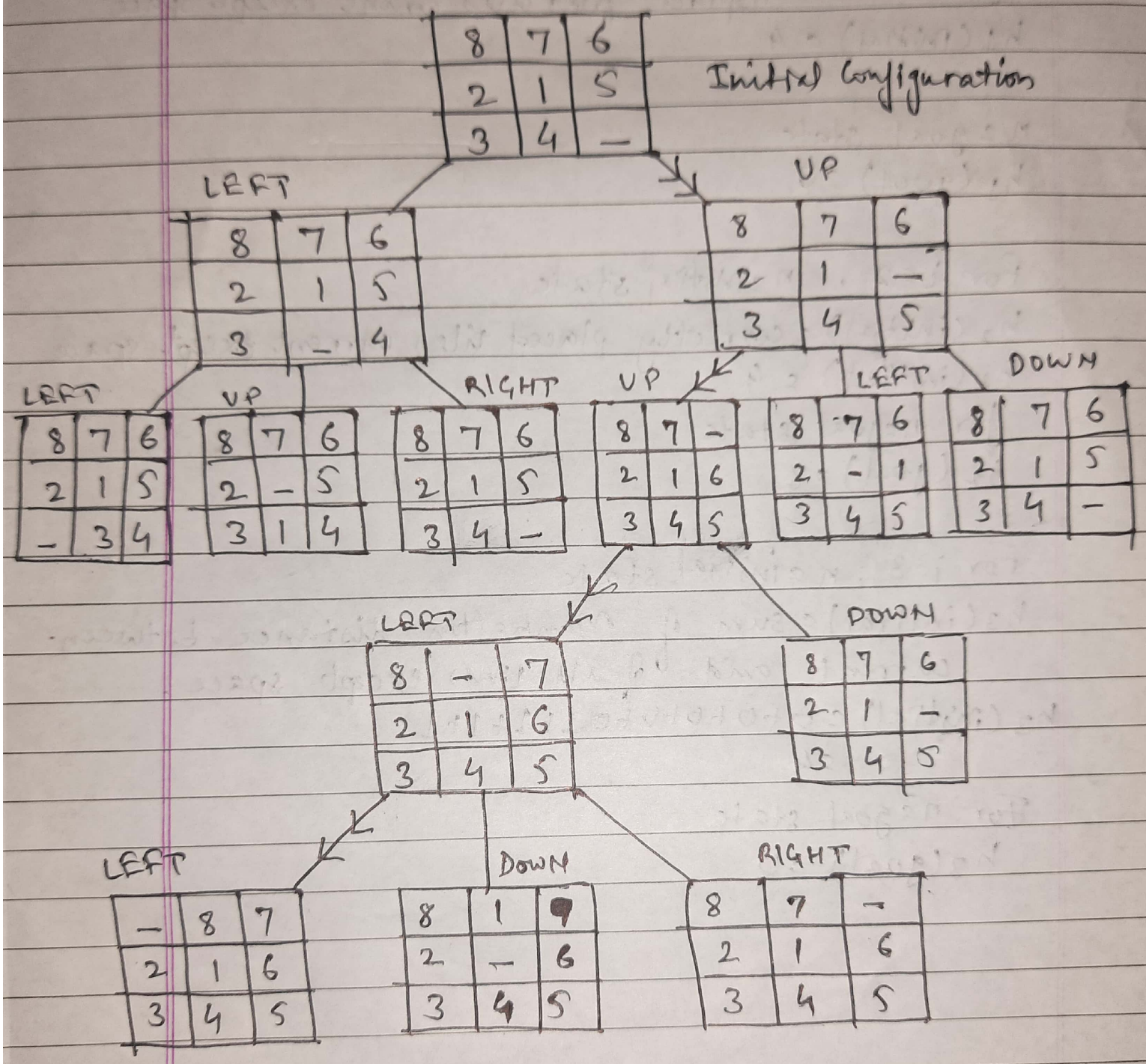
$\{\{8,7,6\},\{2,1,5\},\{-,3,4\}\} \rightarrow \{\{8,7,6\},\{2,1,5\},\{3,-,4\}\} \rightarrow$
$\{\{8,7,6\},\{2,1,5\},\{3,4,-\}\} \rightarrow \{\{8,7,6\},\{2,1,-\},\{3,4,5\}\} \rightarrow$
$\{\{8,7,-\},\{2,1,6\},\{3,4,5\}\} \rightarrow \{\{8,-,7\},\{2,1,6\},\{3,4,5\}\} \rightarrow$
$\{\{-,8,7\},\{2,1,6\},\{3,4,5\}\}$

Since all the moves are equally costly, the cost would be

$$g(n) = 6$$

(c) Draw exhaustive state space tree of depth limited to 4 for instance of 8 puzzle problem in the question. ...(Refer Fig: Q(2))

→



Final Configuration

Page No.
Date

(e) Compute $h_i(n)$ where $i = 1, 2, 3$ & n = initial state, final/goal state from question. (Refer Pg: Q(2))

Ans,

For $i = 1$, n = initial state

h_1 (initial) = Misplace ~~files~~ tiles count except space

h_1 (initial) = 4

n = goal state

h_1 (goal) = 0

For $i = 2$, n = initial state

h_2 (initial) = correctly placed tiles current except space

h_2 (initial) = 4

for n = goal state

h_2 (goal) = 8

For $i = 3$, n = initial state

h_3 (initial) = sum of Manhattan distance between current and of all tiles except space.

h_3 (initial) = 0+0+0+0+1+1+1+1

= 4

For n = goal state

h_3 (goal) = 0