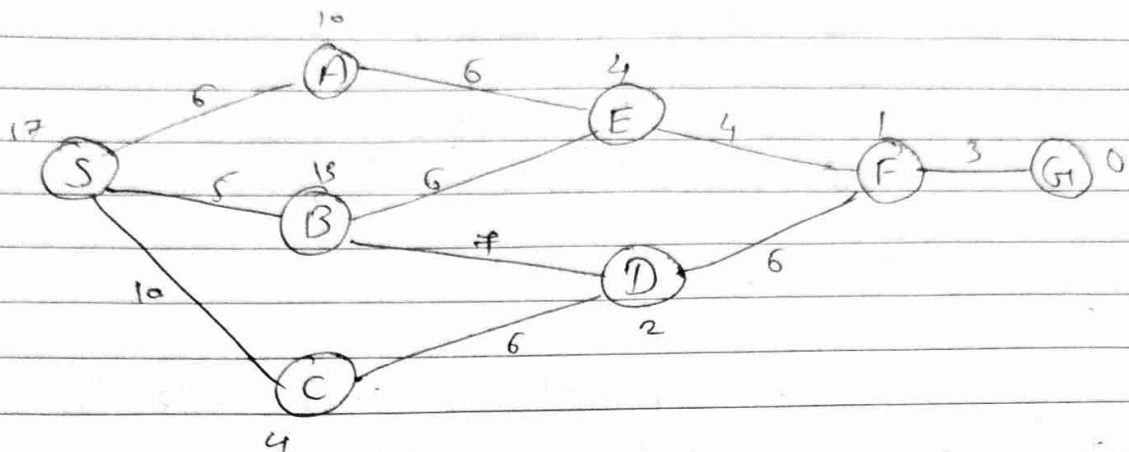Q1 1. Apply BFS on above graph:

BFS is a uniformed search algorithm.
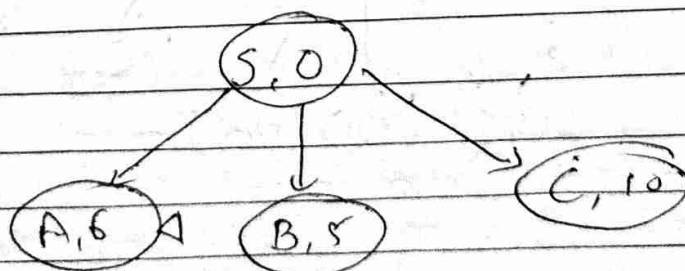
Step 0:
Put initial node S into openlist Path cost for
it is 0.

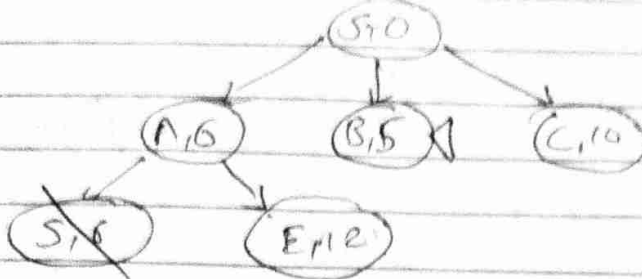(S,0) expand it to generate its child

Step 1:-
Remove S from openlist and since its not goal
node nodes A,B,C.

Step2 :

Remove A as its alphabetically first node in openlist and since its ntgoal node enapand A to generate child nodes S and E.
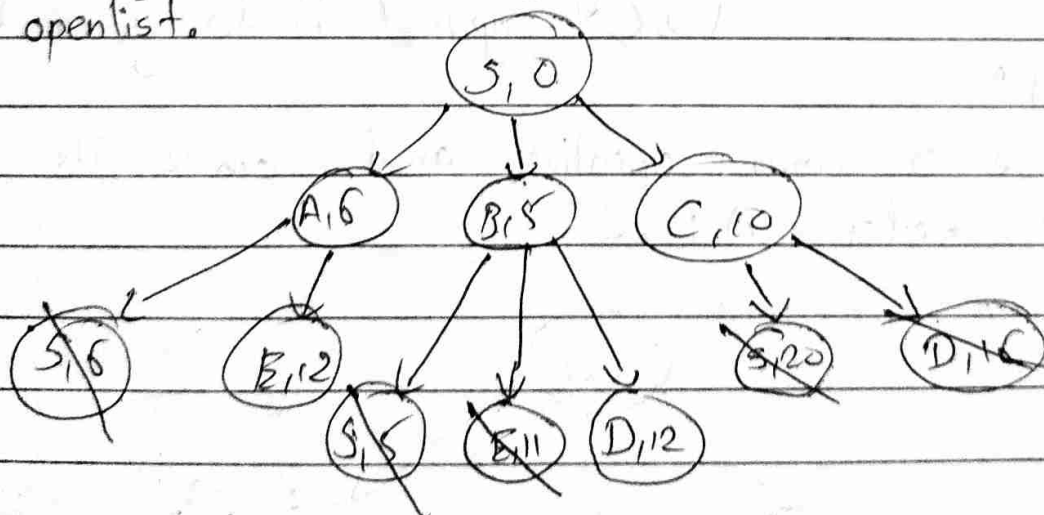

Step 3 :-

Remove B as its alphabetically first node in openlist and since its not goal node expand B to generate child nodes S,E and D.

Step 4 :-

Remove C as its alphabetically first node in openlist & since its not goal node enpand c to generate child nodes S, and D.
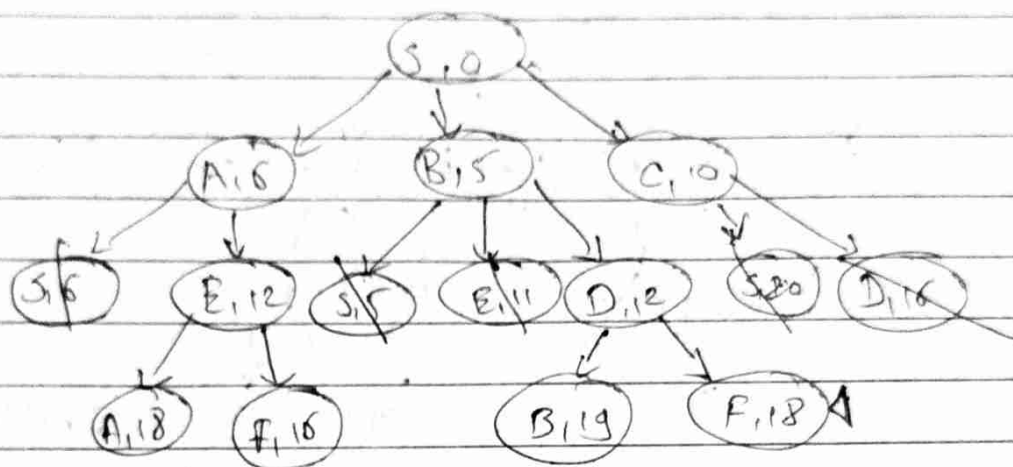
Since they are already in openlist dont put them in openlist.

Step 6 :-

Next we remove E from openlist & since its goal node we will expand it to generate A and F.
Since they are already expanded they will not be put in openlist.

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

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.

S,0
A,6 B,5 C,10
S,6 E,12 S,5 E,11 D,12 S,20 D,16
A,6 F,16 B,19 F,18
D,24 E,22 C,21
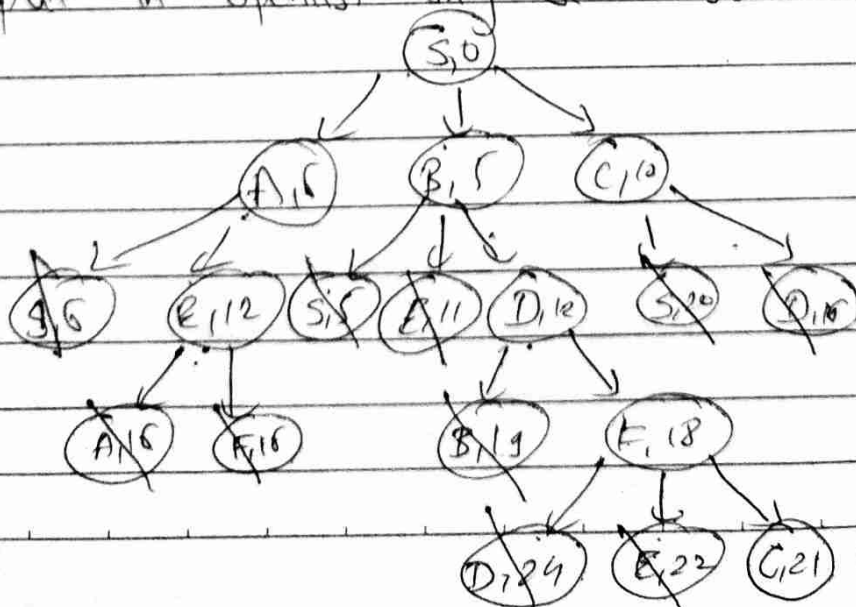
Step 8 :-
Next we remove G from openlist. G is the global
node and hence we stop search & return solution
by traversing back to root node & reversing the
path starting at G.
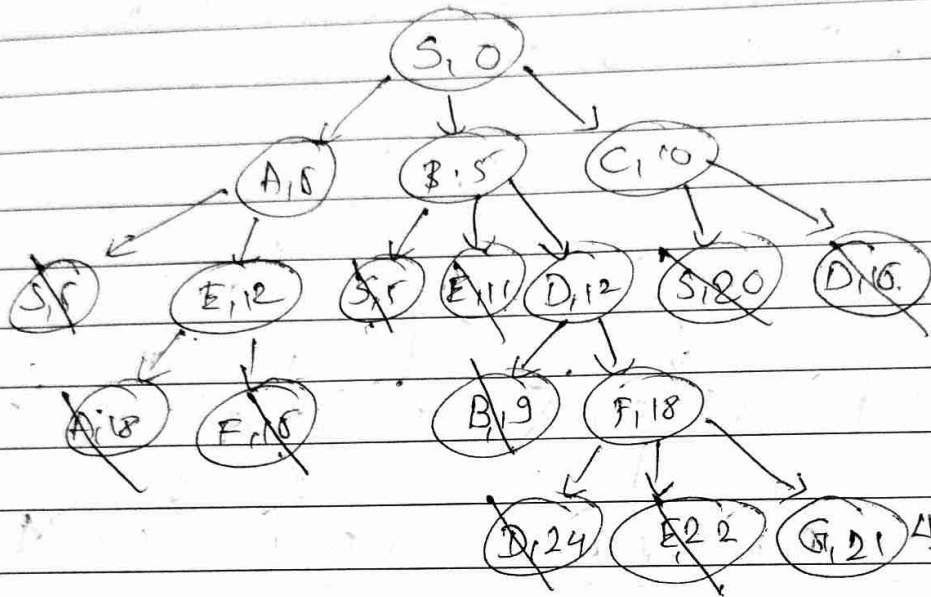
1.4)

Initialization :
  Computer f-source for s o/pt it in openlist.

  f - source $f(s) = h(s) = 17 ((s,17))$

Step 1:-

f - source of successors
$f(A) = h(A) = 10$
$f(B) = h(B) = 13$
$f(c) = h(c) = 4$

(S, 17) → (A, 10) (B,13) (C,4)
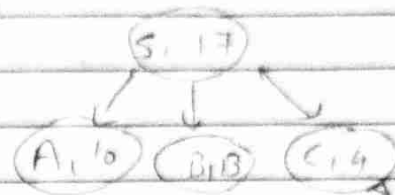
Step 2:-
f - source of successors
$f(s) = h(s) = 17$
$f(D) = h(D) = 2$

(S,17) → (A,10) (B,13) (C,4) → (S,17) (D,7)
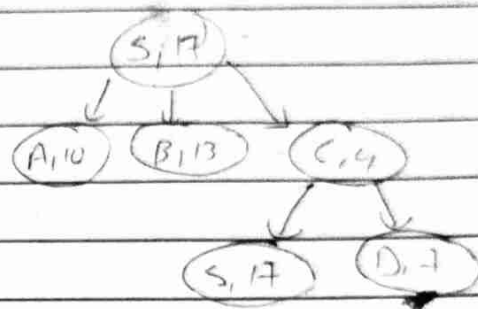
Step 3:-

f - source of successor
$f(c) = h(c) = 4$
$f(B) = h(B) = 13$
$f(F) = h(F) = 1$

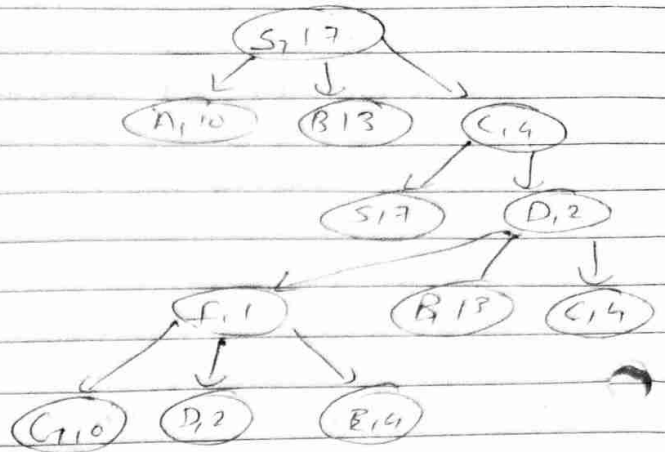(S, 17) → (A,10) (B,13) (C,4) → (S,17) (D,2) → (F,1) (B,3) (...)
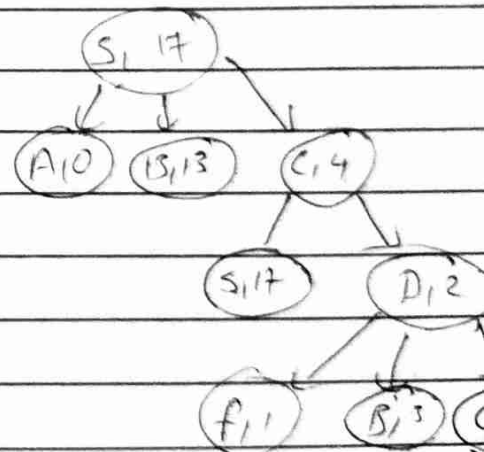
Step 4 :-

f - Source of successor

$f(D) = h(D) = 2$
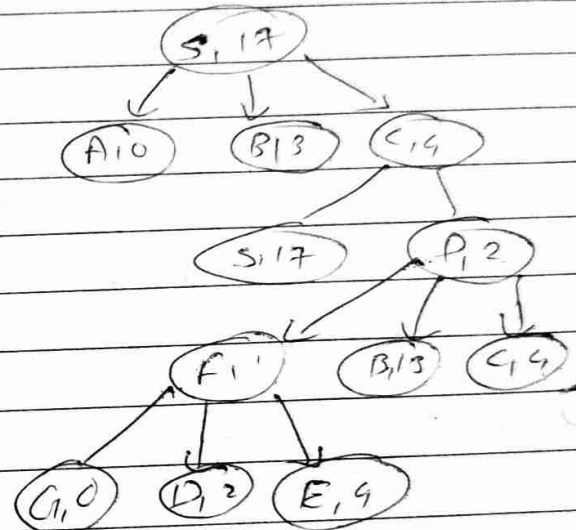
$f(E) = h(E) = 4$

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

Step 5 :-

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

Soln :- $10 + 6 + 6 + 3$

$= 25.$

Q2
a) Lowest path g(n) can be cost to reach goal configuration in least steps. in our case, we can reach final configuration in at least 4 moves.

up, left, left. Since all moves are equally costly we compute g(n) as

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

Consider to'll 8-puzzle instance.

8	7	6
2	1	5
	3	4

Solution can be represented as.

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

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

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

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

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

Since all moves are equally costly cost would
be $g(n) = 6.$

Topic :

c)

$$\begin{array}{|ccc|}
8 & 7 & 6 \\
2 & 1 & 5 \\
3 & 4 & -
\end{array}$$ Initial Config

Left   Up

$$\begin{array}{|ccc|}
8 & 7 & 6 \\
2 & 1 & 5 \\
3 & - & 4
\end{array}$$ $$\begin{array}{|ccc|}
8 & 7 & 6 \\
2 & 1 & - \\
3 & 4 & 5
\end{array}$$

down

Left   Up   right   Up   Left

$$\begin{array}{|ccc|}
8 & 7 & 6 \\
2 & 1 & 5 \\
- & 3 & 4
\end{array}$$ $$\begin{array}{|ccc|}
8 & 7 & 6 \\
2 & - & 5 \\
3 & 1 & 4
\end{array}$$ $$\begin{array}{|ccc|}
8 & 7 & 6 \\
2 & 1 & 5 \\
3 & 4 & -
\end{array}$$ $$\begin{array}{|ccc|}
8 & 7 & - \\
2 & - & 1 \\
3 & 4 & 5
\end{array}$$ $$\begin{array}{|ccc|}
8 & 7 & 6 \\
2 & - & 1 \\
3 & 5 & 5
\end{array}$$ $$\begin{array}{|ccc|}
8 & 7 & 6 \\
2 & 1 & 5 \\
3 & 4 & -
\end{array}$$

$$\begin{array}{|ccc|}
8 & - & 7 \\
2 & 1 & 6 \\
3 & 4 & 5
\end{array}$$ $$\begin{array}{|ccc|}
8 & 7 & 6 \\
2 & 1 & - \\
3 & 4 & 5
\end{array}$$

Left   dow   Right

$$\begin{array}{|ccc|}
- & 8 & 7 \\
2 & 1 & 6 \\
3 & 4 & 5
\end{array}$$ $$\begin{array}{|ccc|}
8 & 1 & 7 \\
2 & - & 6 \\
3 & 4 & 5
\end{array}$$ $$\begin{array}{|ccc|}
8 & 7 & - \\
2 & 1 & 6 \\
3 & 4 & 5
\end{array}$$

Final Configuration.

c)

for i=1   n=initial state
h1 (initial) = misplaces tiles count
                    except space
h2 (initial) = 4

n= goal state
h1 = goal) = 0

for i=2 , n= initial state

h2 (initial) = 4
for n= goal state
h2 (goal) = 8

for i=3 , n= initial state
h3 (initial) = sum of dist. between
          corresp. or correct position of
          all tiles except space
h3 (initial) = 0+ 0+ 0+ 1 +1+ 1+1.
           = 4

for n = goal state
h3 (goal) = 0.