K.G.C.E.
Karjat - Raigad

Assignment - 1A

Page No. :
Date :

Name :- Rohan Shridhar Bhawar

Class :- B.E / I.T

Roll No :- 08

Subject :- IS Lab

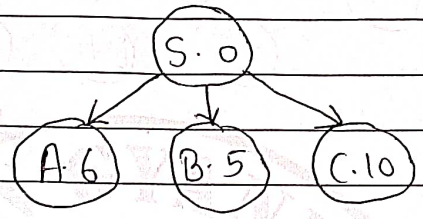
Assignment :- 1A

Q. 1]

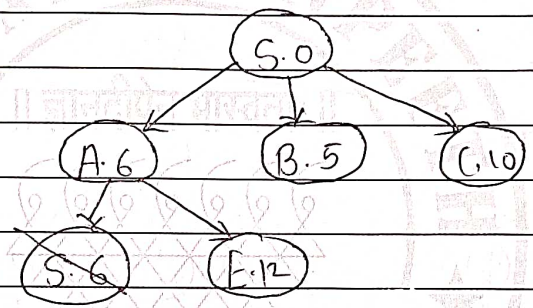
1.1]

⇒ Step 0 :-

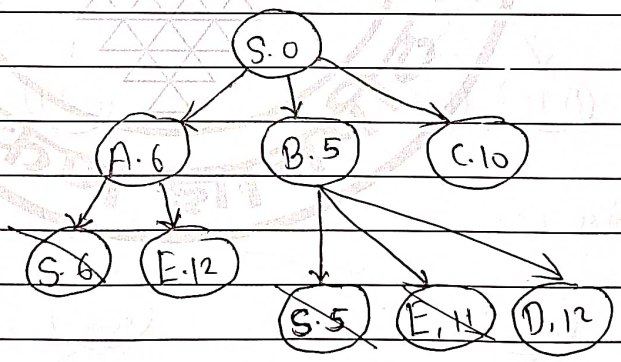
Step 1 :-



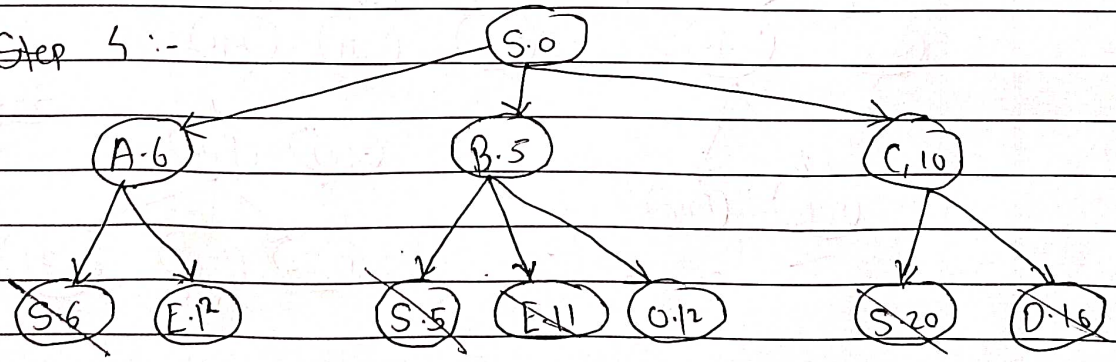
Step 2 :-



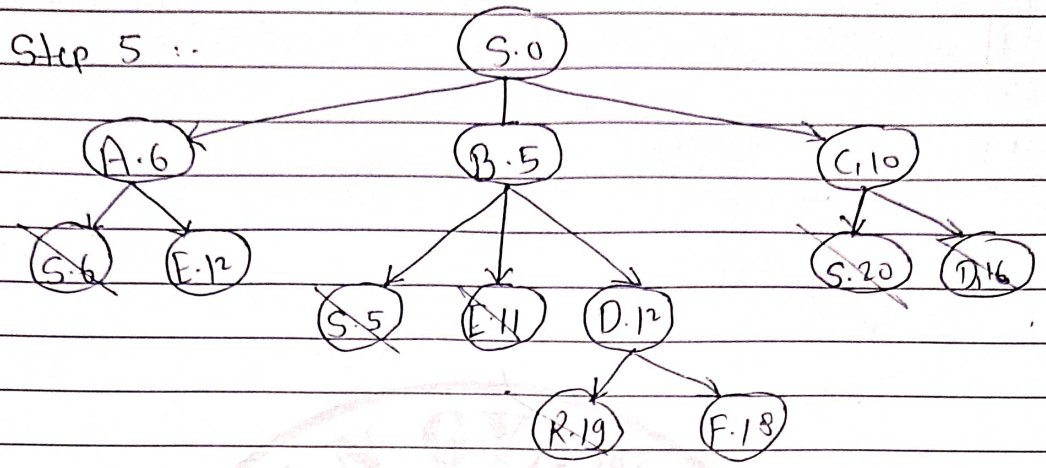
Step 3 :-



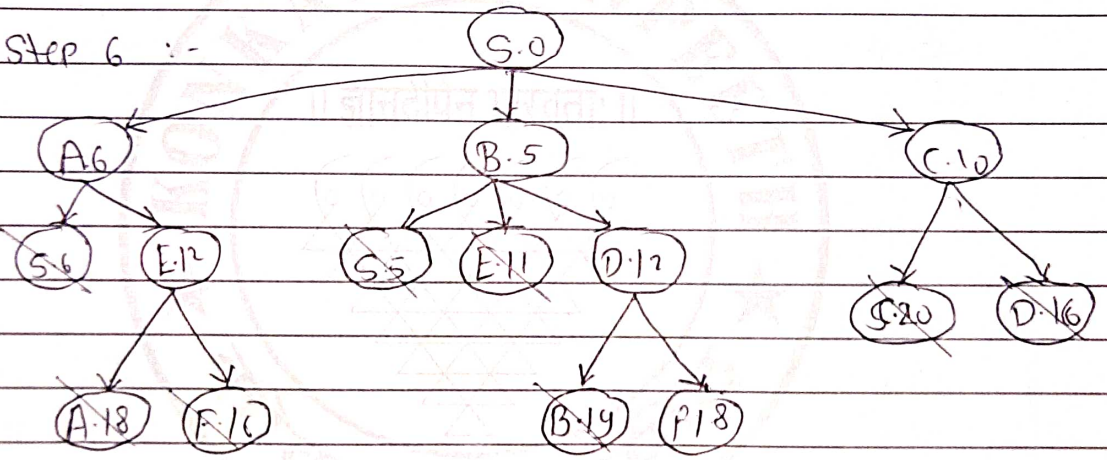
Step 4 :-



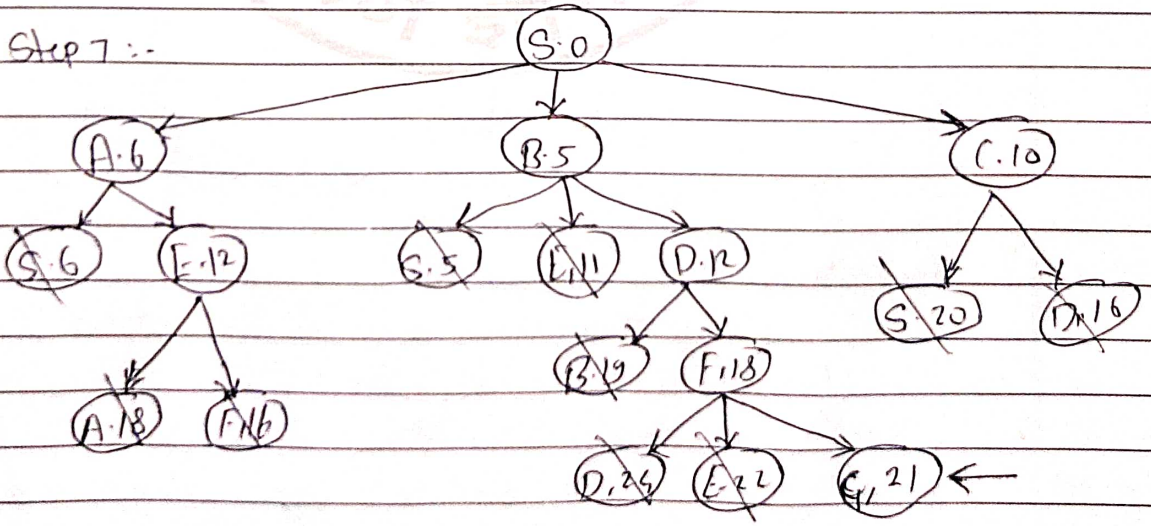
Step 5 :-



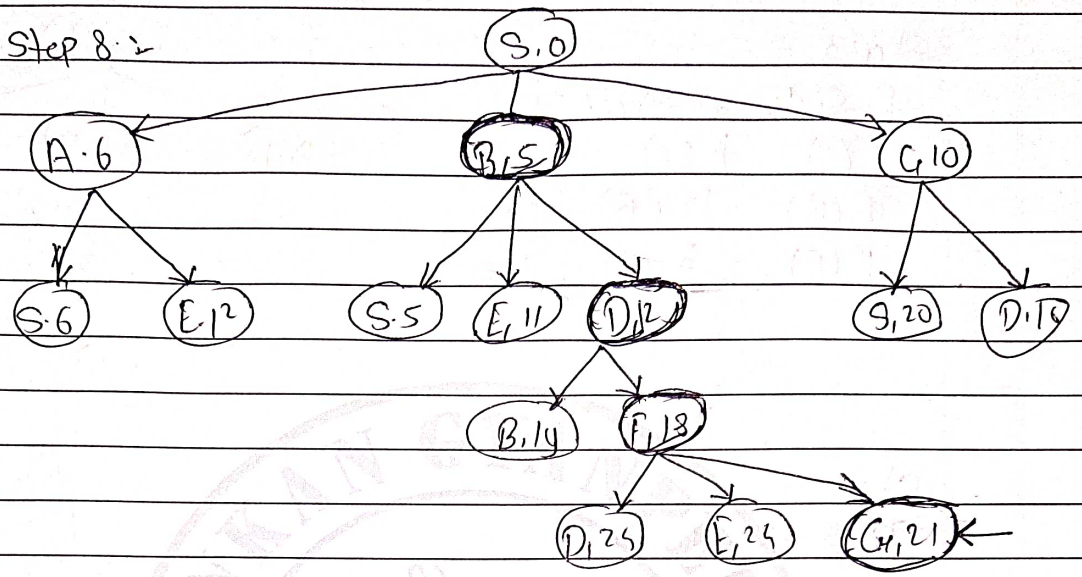
Step 6 :-



Step 7 :-



Step 8 :-



1.4]

⇒ Initialization :- Compute F-Source for S & Put it in openlist

F-Scale S :. $f(S) = h(S) = 17$ (S,17)

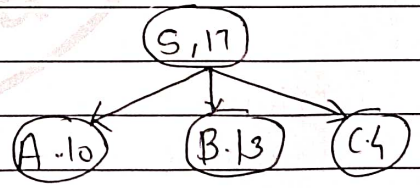
Step 1 :

F - Scale of succes)

$f(A) = h(A) = 10$

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

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

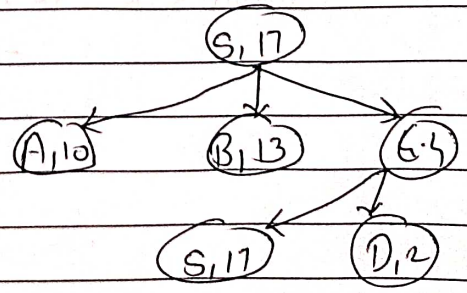


Step 2 :-

P. Scale of succes)

$F(S) = h(S) = 17$

$F(D) = h(D) = 2$



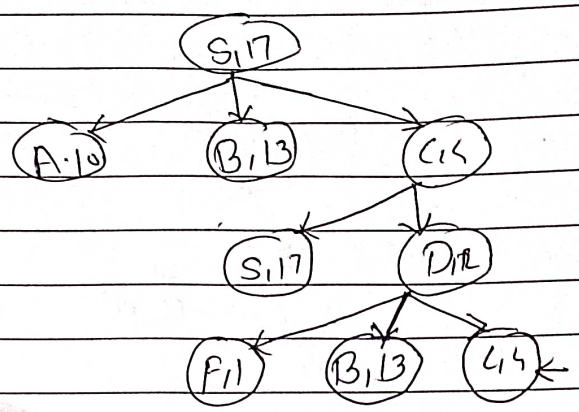
Step 3 :-

F. Scale of S(ulen)

$F(C) = h(C) = 4$

$F(B) = h(B) = 13$

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



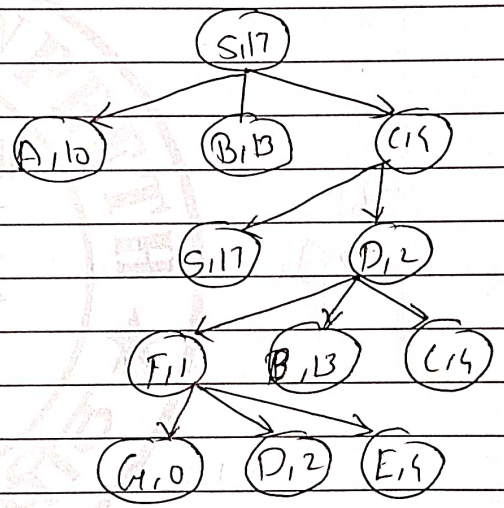
Step 4 :-

F. Scale of S(ulen)

$F(D) = h(D) = 2$

$F(E) = h(E) = 4$

$F(G) = h(G) = 0$



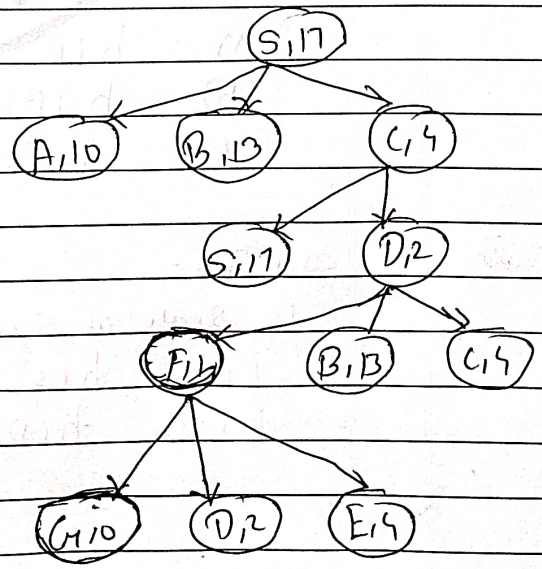
Step 5 :-

Solution is

$S \rightarrow C \rightarrow D \rightarrow F \rightarrow G$ with

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

$= 25$



Q. 2]

a] ⇒ The lowest path cost g(n) can be the ~~cost~~ to reach the goal configuration in least step.

In our case, we can reach the final configuration in at least 4 moves :- UP, UP, Left, Left, since all moves are equally costly we compute g(n) as

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

Consider the following 8-Puzzle instance

8	7	6
2	1	5
-	3	4

Solution can be represented as

{ {8,7,6} {2,1,5} {-,3,4} } → { {8,7,6}, {2,1,5}, {3,-,4} } →

{ {8,7,6}, {2,1,5}, {3,4,-} } → { {8,7,6}, {2,1,-} {3,4,5} } →
{ {8,7,-}, {2,1,5} {3,4,5} } → { {8,-,7}, {2,1,6}, {3,4,5} } →
{ { -,8,7 }, { 2,1,6 }, { 3,4,5 } }

Since all the moves are equally cost will be

$$g(n) = 6$$

c)

8	7	6
2	1	5
3	4	-

Initial config

Left

8	7	6
2	1	5
3	-	4

UP

8	7	6
2	1	-
3	4	5

Left

8	7	6
2	1	5
-	3	4

UP

8	7	6
2	-	5
3	1	4

Right

8	7	-
2	-	1
3	4	5

UP

Left

8	7	6
2	-	1
3	4	5

Left

8	7	6
2	1	5
3	5	-

Right

8	7	6
2	1	5
3	4	-

Left

8	-	7
2	1	6
3	4	5

Down

8	7	6
2	1	-
3	4	5

Left

-	8	7
2	1	6
3	4	5

Down

8	1	7
2	-	6
3	4	5

Right

8	7	-
2	1	6
3	4	5

Final Configuration

e)

⇒ For i=1, n = initial state

h1 (initial) = Misplaced tiles count except space

h1 (initial) = 4

n = goal state

h1 (goal) = 0

For i = 2. n = initial state

h2 (initial) = exactly replaced tiles count except space

h2 (initial) = 4

For n = goal state

h2 (goal) = 8

For i = 3, n = initial state

h3 (initial) = sum of manhattan dist betw^n current and correct position of all tiles except space

h3 (initial) = 0 + 0 + 0 + 0 + 1 + 1 + 1 + 1

= 4

For n = goal state

~~h(3)~~ h3 (goal) = 0