K.G.C.E.
Karjat - Raigad

Page No. :
Date :

KGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCE

:> Name : Tejas Balaso Jadhav.

:> Class :- BE - IT.

:> Roll No. :- 24

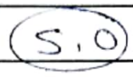
:> Subject :- IS Lab

D.O.P. D.O.A. Remark Sign

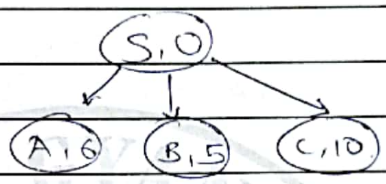
Q. 1

1.1

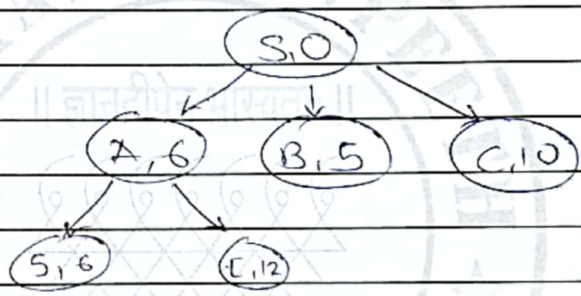
Step 0:



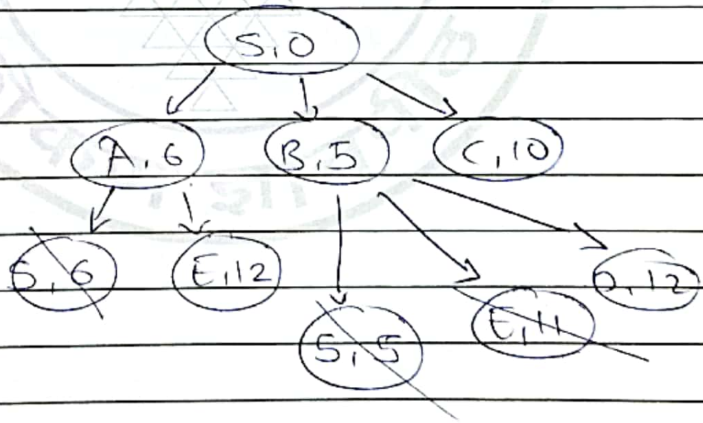
Step 1:



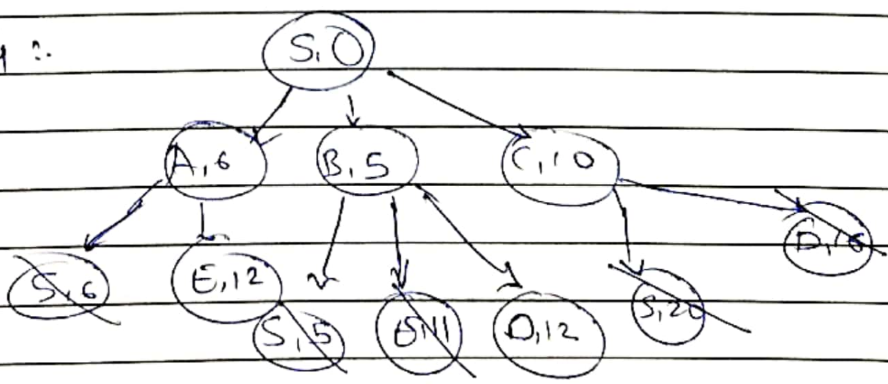
Step 2:



Step 3:

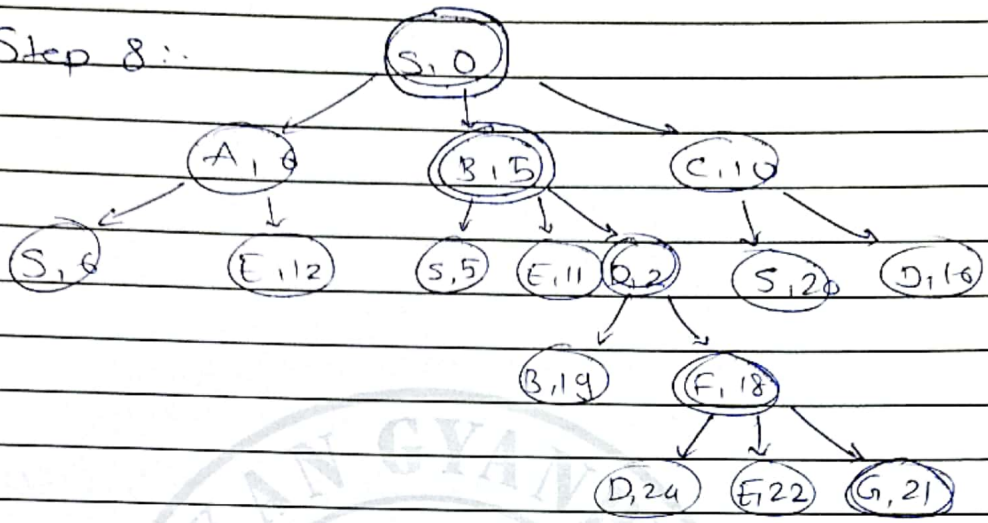


Step 4:



KGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCE

Step 8 :-



1.4

→

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

F-score S : $f(s) = h(s) = 17$ (S, 17)

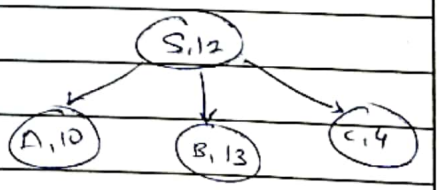
Step 1 :

F = score of successor

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

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

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

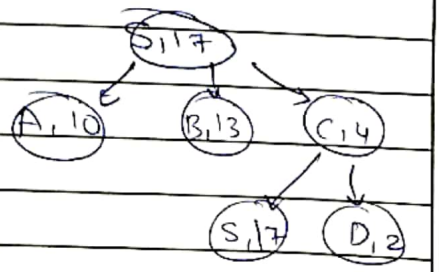


Step 2 :

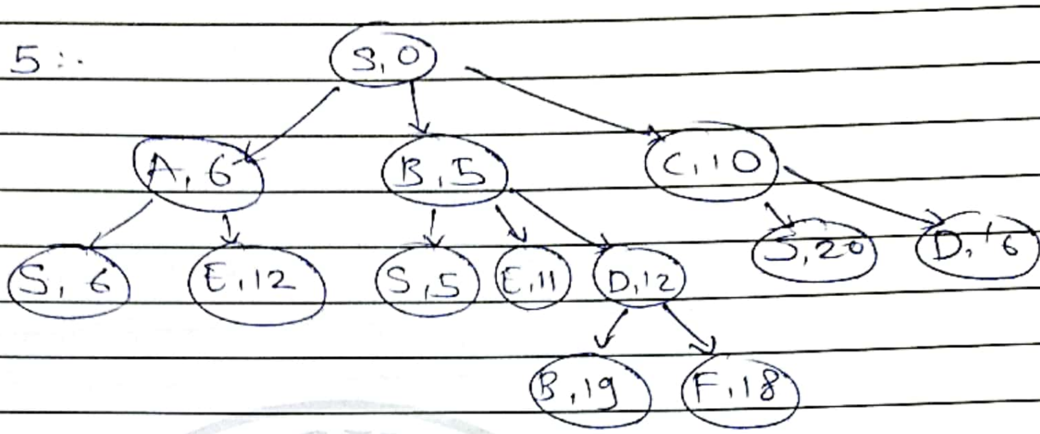
F - score of successors

$f(S) = h(S) = 17$

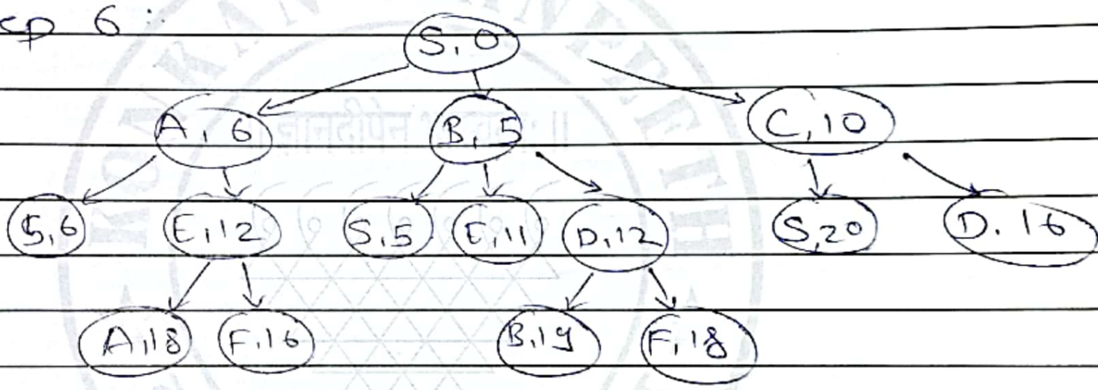
$f(D) = h(D) = 2.$



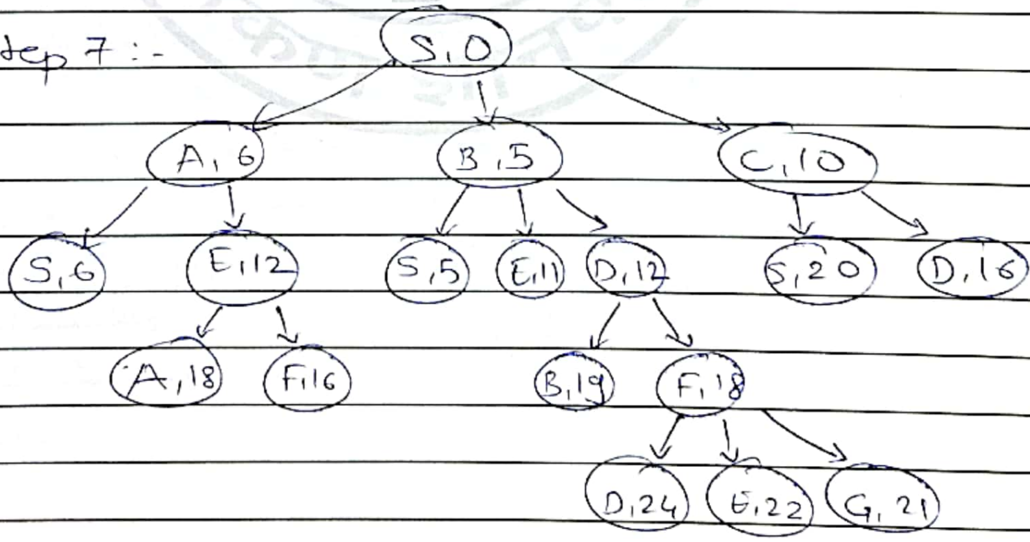
Step 5 :-



Step 6 :-



Step 7 :-



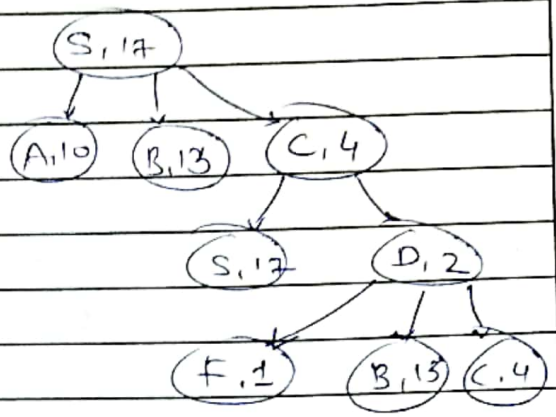
Step 3:

f = score of successor

f(C) = h(C) = 4

f(B) = h(B) = 13

f(F) = h(F) = 1



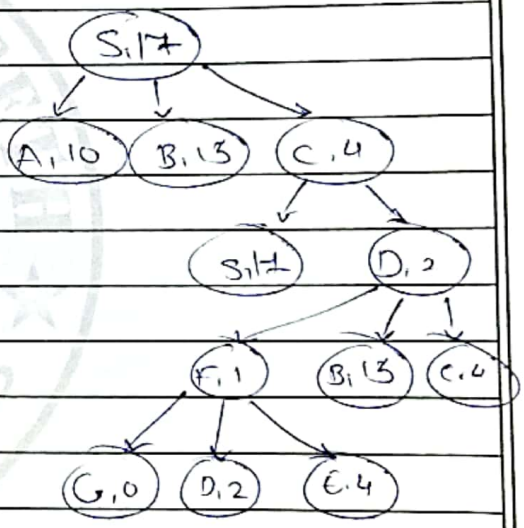
Step 4:

f = score of successor

f(D) = h(D) = 2

f(E) = h(E) = 4

f(G) = h(G) = 0



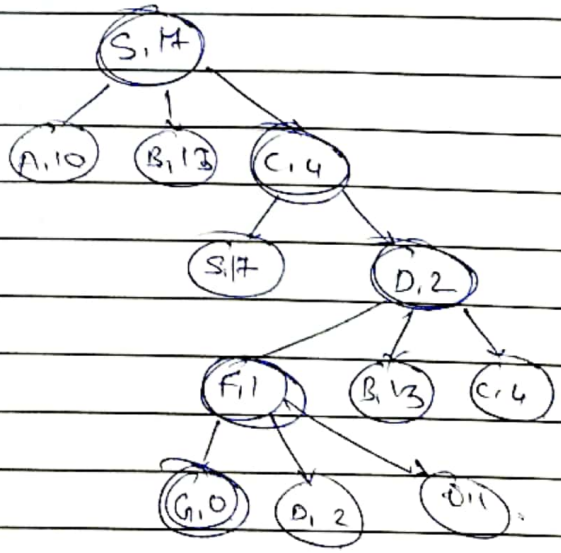
Step 5:

Solution is :-

S → C → D → F → G with

Solution cost : 10 + 6 + 6 + 3

= 25



Q. 2

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

In our case, we can reach the final 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 + 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} } →

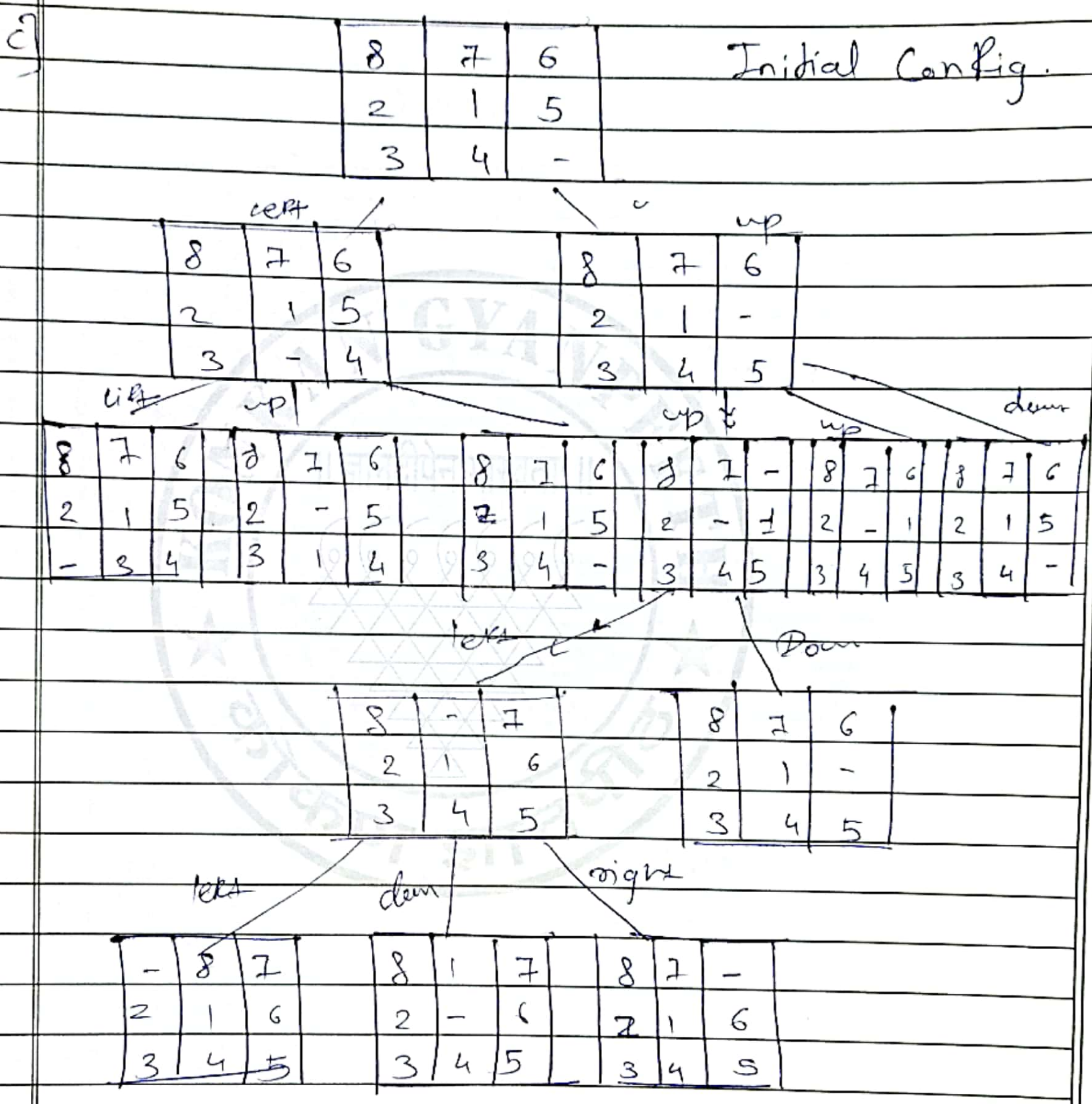
{ {8,7,-} {2,1,5} {3,4,5} } → { {8,-,7} {2,1,6} {3,4,5} } →

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

Since all the moves are equally costly the cost would be

g(n) = 6

5. Path :



Final Configuration.

6] For i = 1, n = initial state.
h1(initial) = Misplaced tiles count except space.
h2(initial) = 4

n = goal state.
h1(goal) = 0.

For i = 2, n = initial state.
h2(initial) = Correctly explaced 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. between current & correct position of all tiles except space.
h3(initial) = 0 + 0 + 0 + 0 + 1 + 1 + 1 + 1
= 4

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