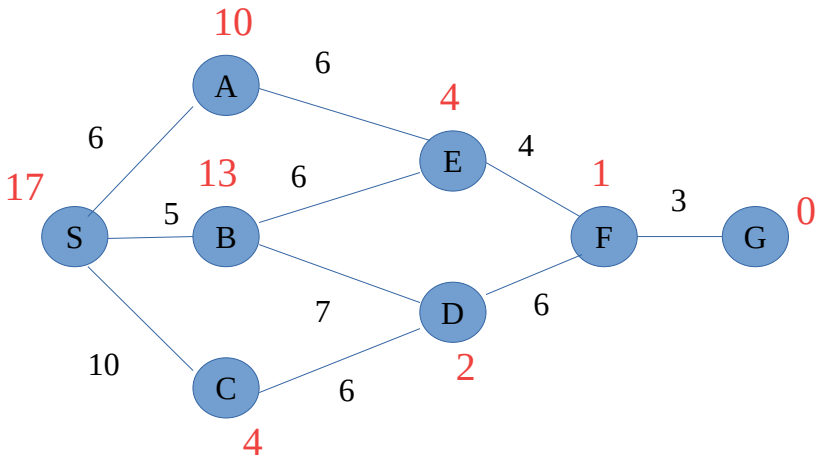
Assignment 1

Subject: AI Date: 30/10/2021

Part A

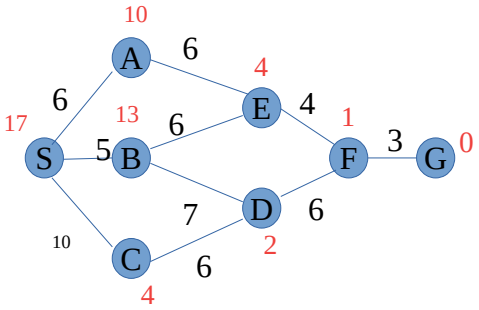
Note:
1. To be completed by 30/10/21 in writting and Upload the answer writeup on kgcelms.in
2. Submission link will be active for submission from 10th Sep 2021 to 20th Nov 2021.

Q1. 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 the hueristic function value.



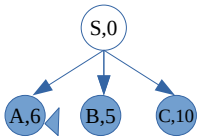
Using above data solve following problems:

Q1.1. Apply BFS on above graph

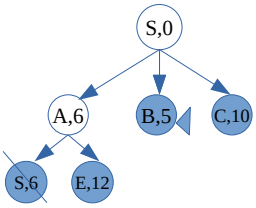


BFS (Breadth First Search is a Uninformed search algorithm) we will show steps in Gaph Search implementation of this algorithm on above problems:
Step 0: Put initial node S into openlist Path cost for it is 0.
Step 1: Remove S from openlist and since its not goal node 

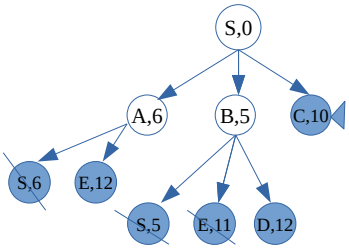
expand it to generate its child nodes A, B, C



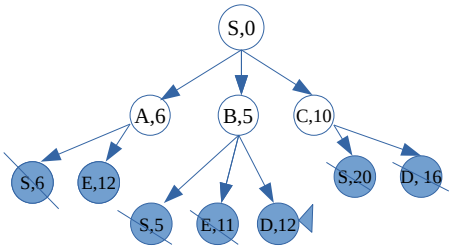
Step 2: Remove A as its alphabetically first node in openlist and since its notgoal node exapand A to generate child nodes S and E



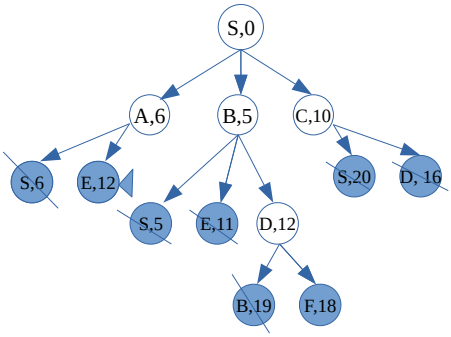
Step 3: Remove B as its alphabetically first node in openlist and since its not goal node exapand B to generate child nodes S, E and D.



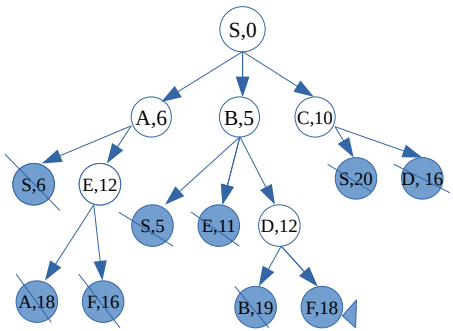
Step 4: Remove C as its alphabetically first node in openlist and since its not goal node exapand C to generate child nodes S, and D. Since they are already in openlist dont put them in openlist.



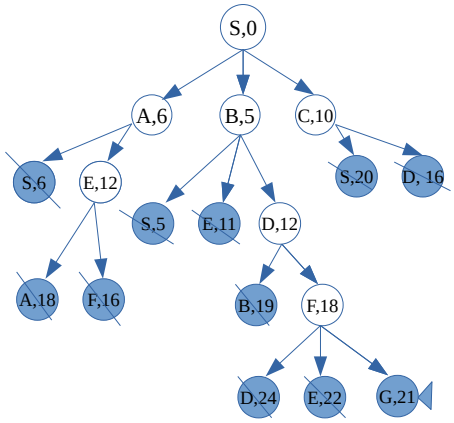
Step 5: Level 1 is completely traversed and hence we move to level 2. We remove D from openlist and since its goal node we will expand it to generate B and F



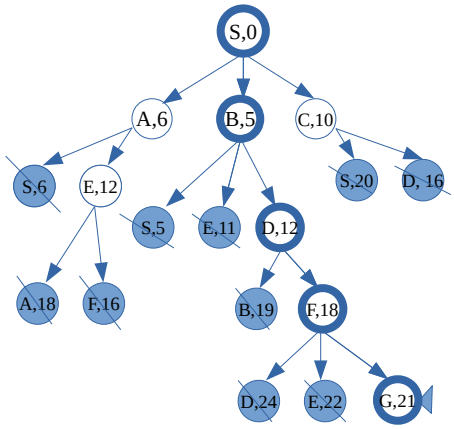
Step 6: Next we remove E from openlist and since its goal node we will expand it to generate A and F. Since they are already expanded they will not be put in openlist.



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



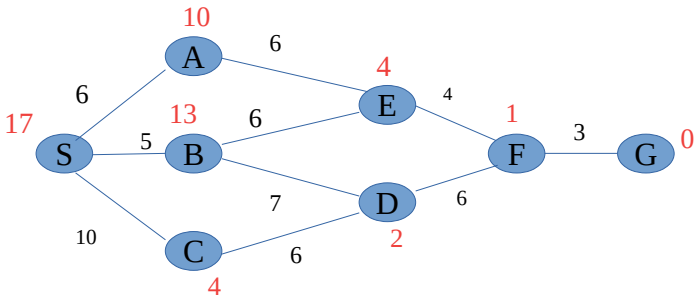
Step 8: Next we remove G From openlist. G is the goal node and hence we stop search and return solution by traversing beck to root node and reversing the path starting at G.



Q1.2. Apply Depth Limited DFS with l=3. For this algorithm to be complete what should be the minimum value of l.

Q1.3 Apply Uniform cost search on the problem above

Q 1.4. Appy Best First Search and clearly show all the steps using search tree.

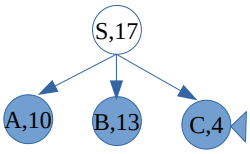


Initialization : Compute f-score for S and put it in the openlist.

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

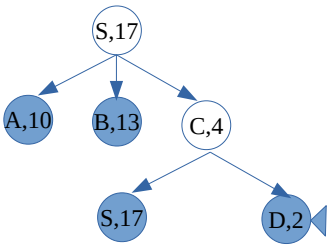
Step 1: Remove S from openlist and since its not goal node exapand it. For each successor A, B, C compute f-score and put them in open list with increasing order of f-score.

F-score Of Successors
f(A) = h(A) = 10
f(B) = h(B) = 13
f(C)= h(C) = 4



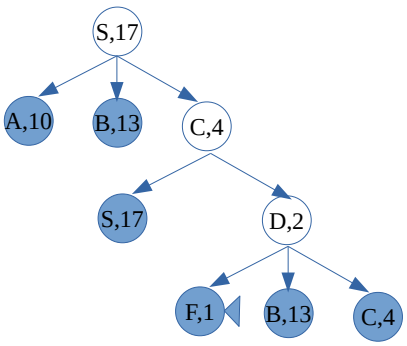
Step 2: Remove C from openlist and since its not goal node exapand it. For each successor S, D compute f-score and put them in open list with increasing order of f-score.

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



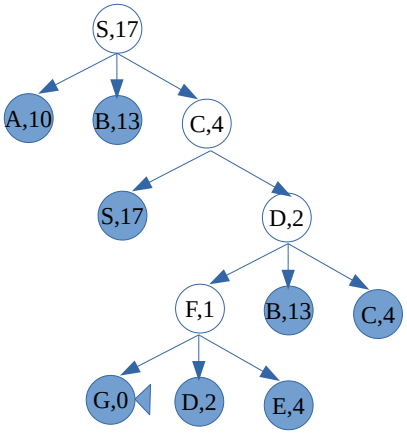
Step 3: Remove D from openlist and since its not goal node exapand it. For each successor C, B, F compute f-score and put them in open list with increasing order of f-score.

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 and since its not goal node exapand it. For each successor D, E, G compute f-score and put them in open list with increasing order of f-score.

F-score Of Successors
f(D) = h(D) = 2
f(E) = h(E) = 4
f(G) = h(G) = 0

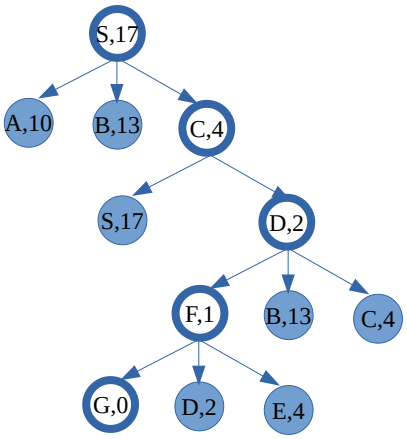


Step 5 Remove G from openlist and since its goal node it passes goal test. Stop search and traversebackwards following paret pointers to the root node of search tree, reverse the path traversed return it as solution to problem.

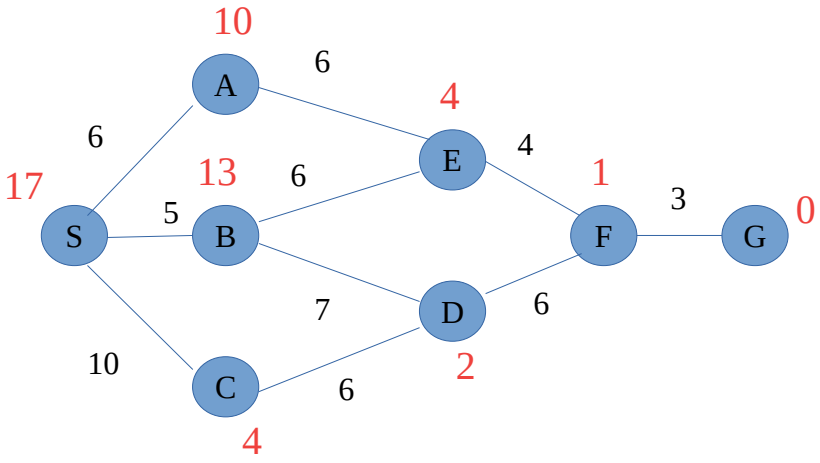
Solution is S -> C -> D -> F -> G with

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

This is solution is not an optimal solution.

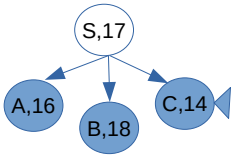


Q 1.5. Appy A* algorithm and clearly show all the steps using search tree.



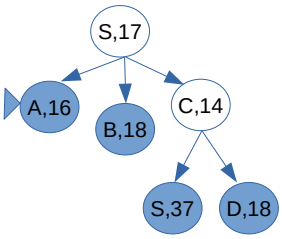
Step 1: calculate f-score for S (f(S)=0+17=17) put S in open list open list.

Step 2: get S out of openlist and since its not goal apply sucessor function to get A, B and C as its successors. Compute f-score for them and put them in openlist in priority order.



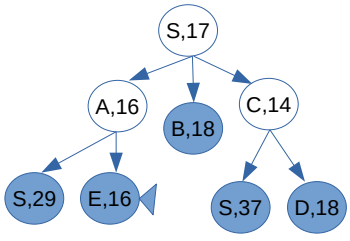
f(A) = 6 + 10 =16
f(B) = 5 + 13 =18
f(C) = 10 + 4 = 14

Step 3: get C out of open list and since its not goal apply sucessor function to get S, D as its successors. Compute f-score for them and put them in openlist in priority order.

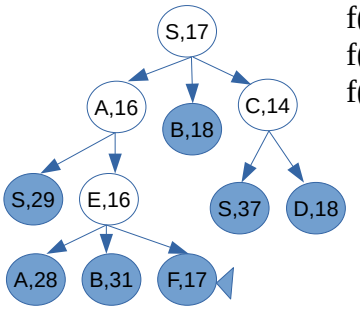


f(S) = (6+6) +17 = 29
f(E) = (6+6) + 4 = 16

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

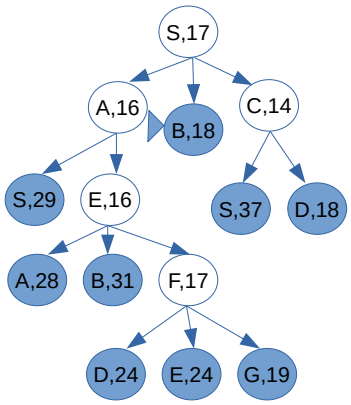


Step 5: get E out of open list and since its not goal apply sucessor function to get A, B and F as its successors. Compute f-score for them and put them in openlist in priority order.

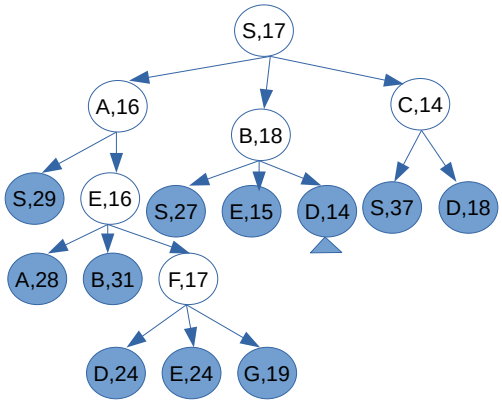


f(A) = (6+6+6) +10 =28
f(B) = (6+6+6) + 13 = 31
f(F) = (6+6+4) + 1 =17

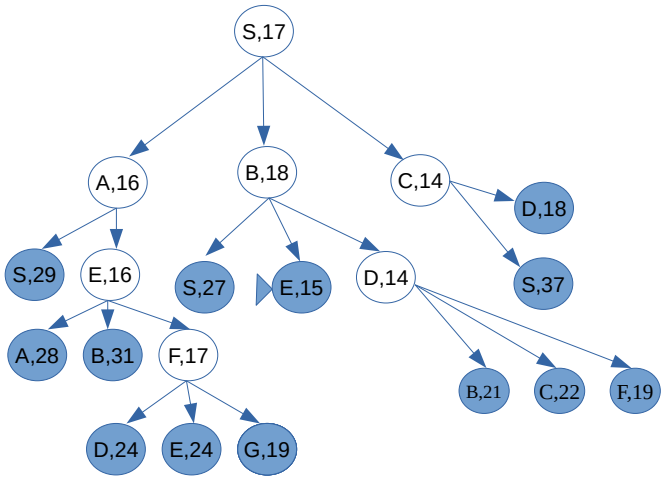
Step 6: get F out of open list and since its not goal apply sucessor function to get D, E and G as its successors. Compute f-score for them and put them in openlist in priority order.



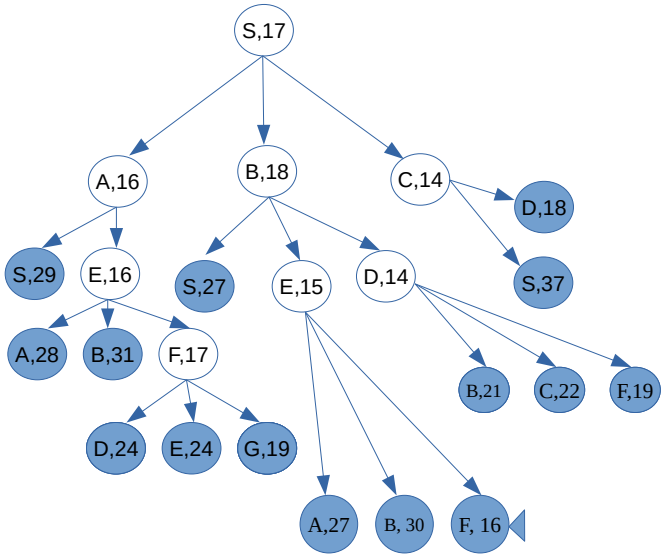
Step 7:



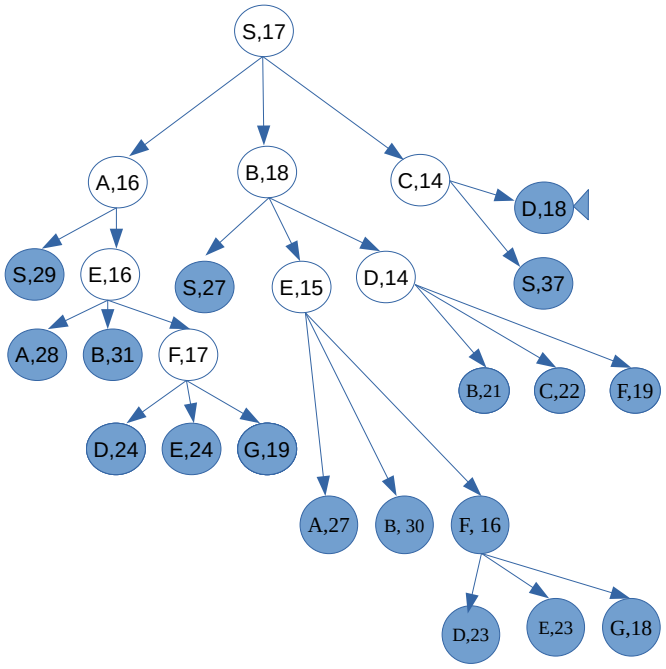
Step 8



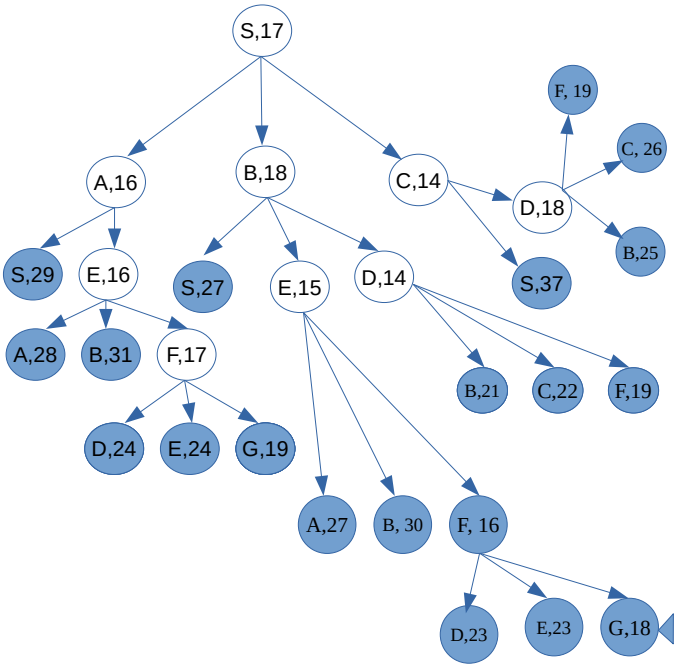
Step 9:



Step 10:

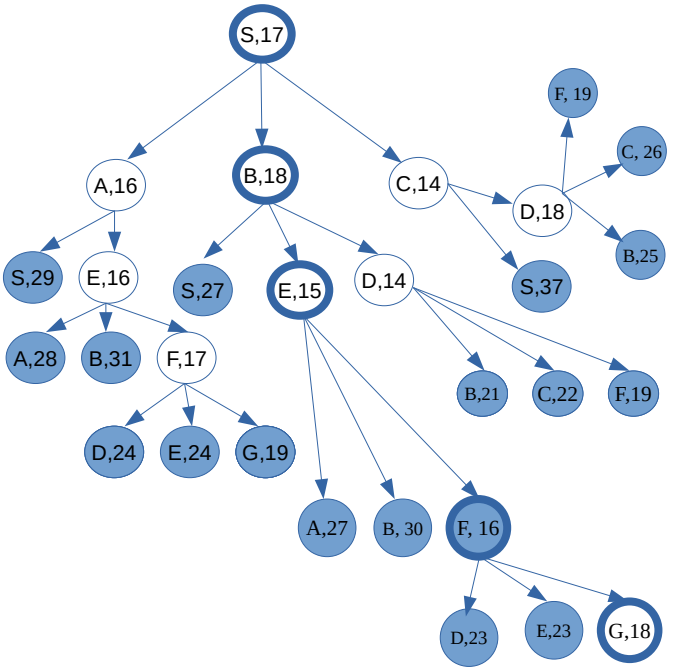


Step 11:

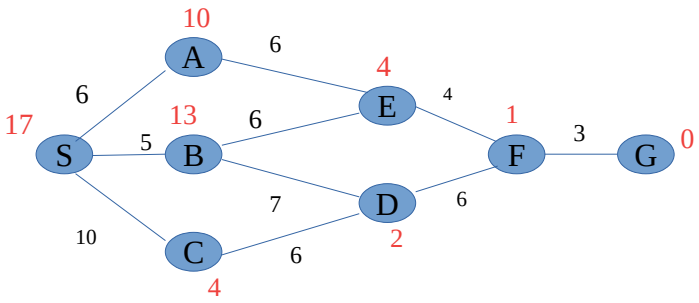


Step 12 : Remove G,18 from open list, Since G is goal node goal test results in true and we will find the solution and return it. We traverse backwards towards the root from G, 18 and reverse the path we traverse to get the solution.

Hence Solution is **S ->B -> E -> F -> G with path cost 18**. This is the optimal path cost Final Search Tree after A * will be:

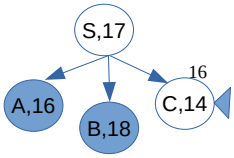


Extra Question Same problem with RBFS



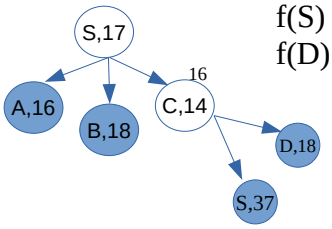
Step 0: Call RBFS with S as initial node and f-limit as infinity.

Step 1: RBFS(problem, S, infinity)
a. Since S is not goal node. Generate its successors A, B and C Compute f-score for them.
b. if new f-score is > than earlier score (those visited earlier will have f-score computed) or f-score initialised first time then f-score for that successor is updated.
c. best is set as C and alternative as A
d. update f_limit to C.f-score = 14.
e. call RBFS with problem definition, C and min(14,16)



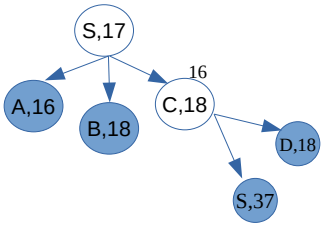
f(A) = g(A)+h(A)=16
f(B) = g(B)+h(B)=18
f(C) = g(C)+h(C)=14

e.a Since C is not goal node. Generate its successors S and D Compute f-score for them.
e.b update f-score of S and D as 37 and 18 respectively
e.c best is set as D and alternate as S
e.d return failure and best.f=18 since best.f > f_limit



f(S) = (10+10) + 17 =37 since 37 > 17 update S f-score
f(D) = (10+6) + 2 =18

f update C.fscore as 18 returned from call RBFS with problem definition, C and min(14,16)



g. Now A is best node and C as alternate node

Note: Complete this further.....

Q2. Consider 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

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 wihin least number of steps. All moevs are thus equally costly. Define g(n) in you own words. What will be the cost of 6 step solution to some arbitrary 8 puzzle instance ?

b. Does any arbitrary instance of 8 puzzle represented as state space problem, has single solution ?
If yes mention the solution for above configuration of initial and goal sates ?
If no write atleast two distinct solutions for above configuration of initial and goal sates ?
In the solution/s also mention at which depth level the goal was found.

c. Draw exhaustive state space tree of depth limited to 4 for instance of 8 puzzle problem in the question.

d. In the tree above name the nodes with letters A, B, C ... starting at leaf level from left to right in horizontal scan line up to initial node. Now for following search strategies mention sequence of visited nodes untill the tree is fully visited.
i. Breadth First Search ii. Depth First Search iii. Iterative deepening Depth First Search

e. Compute $h_i(n)$ where i=1,2,3 and n = initial state, goal state from question.

f. Apply Greedy Best First Search to the problem with h1 as hueristic function and g(n) you defined in part a. Show clearly computation of evaluation function and search steps. Show steps up to generation of one node at depth level 4 or you reach the solution which ever is early.

g Apply A^* Search to the problem with h1 as hueristic function and g(n) you defined in part a. Show clearly computation of evaluation function and search steps. Show steps up to generation of one node at depth level 4 or you reach the solution which ever is early.

Q 3. Consider following formulation of water jug problem:

1. State definition : <x,y> ---> indicates x lit water in jug with capacity 5 and y lit water in jug with capacity 3 lit
2. Initial state : <0,0> ---> both jugs are empty
3. Goal states : <4,0>, <4,1>, <4,2> ---> 5 lit jug has 2 lit and 3 lit jug has upto 2 lit of water
4. Acions allowed:
    1. <x,_> ---> <0,_> We just empty jug 1
    2. <_,y> ---> <_,0> We just empty jug 2
    2. <x,_> ---> <5,_> We just fill jug 1 to its capacity
    4. <_,y> ---> <_,3> We just fill jug 2 to its capacity
    5. <x,y> ---> <x-z,3> We fill jug 2 to its capacity by transferring water from +jug 1 here **y+z=3**
    6. <x,y> ---> <5,y-z> We fill jug 1 to its capacity by transferring water from jug 2 here x**+z=5**
5. Step cost will be equal for all actions and one step counts as 1 unit cost.

Based on above problem formulation answer following questions.

a. Draw fully expanded search tree if you are doing eqhaustive search up to three levels for specifications given

b. How many goal nodes are present in above search tree mention their respective ddepth.
c. What is the depth of shallowest goal node also mention the problem state for same.

d. Apply Breadth First Search for above problem and find out the sequence of actions in the solution. How many nodes and in which order are visited before reaching the goal node.

e. Apply Depth First Search for above problem and find out the sequence of actions in the solution. How many nodes and in which order are visited before reaching the goal node.

f. What should be minimum value of depth limit set by the algorithm so that Depth Limited Depth First Search when applied to this problem gives you optimal solution. Explain how you arrive at this value.