Best First Search

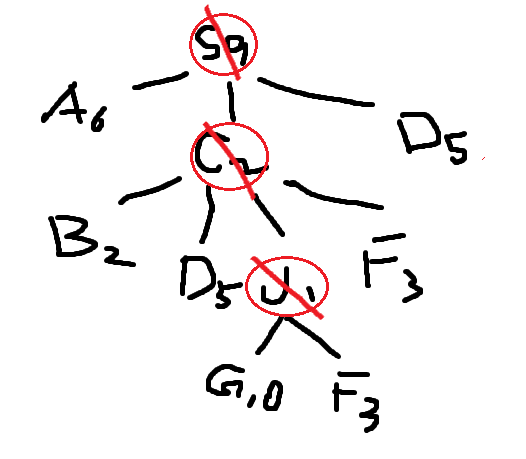
Greedy Search f(n)=h(n)

A\*Search f(n)=g(n)+h(n)

Greedy Search: f = h

hSLD(n):the direct distance from n to target node

Calculate each action from F, find the lowest evaluation one(heuristic), make it be the best action.



Open = [S9]; closed = [];

Open = [C2, D5, A6]; closed = [S9];

Open = [J1, B2, F3, D5, A6]; closed = [S9,C2];

Open = [G1o, B2, F3, D5, A6]; closed = [S9,C2, J1];

S9— A6 — G1o

S9—  C2 — J1— G1o

                                 J1— G1o

                C2— B2 — G1o

                C2— F3 — G2o

                C2— D5 — E5 — G2o

S9— D5 — E5 —G2o



Best first searches the node for the lowest value until its searches the last node of the branch, it will continue the search until it has found the node it is looking for. In our implementation it starts from the node ‘S’ and finishes at the node ‘G1’. From the tree diagram it will search for the lowest node which in this case is ‘C’ it will then continue the search and move down a branch and search for the next value which will be ‘J’ the next node will be ‘G1’ as the cost is 0 and so the program will stop as it has found the node.