1)best-first search

graph={

"A":({"B":9,"C":4,"D":7},21),

"C":({"A":4,"E":17,"F":12},18),

"B":({"A":9,"E":11},14),

"E":({"B":11,"C":17,"Z":5},5),

"Z":({"E":5,"F":9},0),

"F":({"C":12,"D":14,"Z":9},8),

"D":({"A":7,"F":14},18)

}

def greedy\_search(graph,prev,dst,path,q):

print("Connected nodes of current node",prev,"with h(n) values: ")

for n in graph[prev][0]:

if n not in path:

q[n]=graph[n][1]

print(n,"->",q[n])

while q:

mn=min(q,key=q.get)

print("Taking minimum h(n) vertext: ",mn)

q.pop(mn)

if dst==mn:

return path+[dst]

new\_path=greedy\_search(graph,mn,dst,path+[mn],q)

if new\_path:

return new\_path

return []

result = greedy\_search(graph, "A", "Z", ["A"], {})

print("Greedy search recurssive: ","->" .join(result)if result else "Path not found!")

output:

Connected nodes of current node A with h(n) values:

B -> 14

C -> 18

D -> 18

Taking minimum h(n) vertext: B

Connected nodes of current node B with h(n) values:

E -> 5

Taking minimum h(n) vertext: E

Connected nodes of current node E with h(n) values:

C -> 18

Z -> 0

Taking minimum h(n) vertext: Z

Greedy search recurssive: A->B->E->Z