**Facultad de Ingeniería en Electricidad y Computación Artificial Intelligence**

**GROUP # 8**

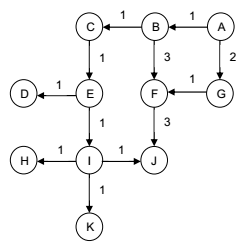
* Iván Mosquera
* Kleber Diaz
* Edgar Carvajal

**Homework # 2.** Heuristic Search Mechanisms:

**Work as a group** and answer each of the following questions. You must report only onedocument as a group, and place it in SIDWeb 4.0, section “Trabajos”.

***DO NOT*** *send the homework via email.*

1. Consider the following graph, where nodes are labelled alphabetically and links have associated costs. As a group discuss and answer the following questions?



* 1. List the nodes in the order they would be visited when performing depth first search for *K*, starting from *A*. Assume that neighbors of the same node are visited in alphabetical order.

1. OPEN = [A]; CLOSED=[] ; X =[]
2. OPEN=[B,G] ; CLOSED =[A] ; X =[A]
3. OPEN=[C,F,G] ; CLOSED =[B,A]; X =[B]
4. OPEN=[E,F,G] ; CLOSED =[C,B,A]; X =[C]
5. OPEN=[D,I,F,G] ; CLOSED =[E,C,B,A]; X =[E]
6. OPEN=[I,F,G] ; CLOSED =[D,E,C,B,A]; X =[D]
7. OPEN=[H,J,K,F,G] ; CLOSED =[I,D,E,C,B,A]; X =[I]
8. OPEN=[J,K,F,G] ; CLOSED =[H,I,D,E,C,B,A]; X =[H]
9. OPEN=[K,F,G] ; CLOSED =[J,H,I,D,E,C,B,A]; X =[J]
10. OPEN=[F,G] ; CLOSED =[J,H,I,D,E,C,B,A]; X =[K] **“K” WAS FOUND.**

ORDER : A,B,C,E,D,I,H,J,”K”

* 1. List the same list for breath first search for *K*, starting from *A*.

1. OPEN=[A] ; CLOSED =[] ; X =[]
2. OPEN=[B,G] ; CLOSED =[A] ; X =[A]
3. OPEN=[G,C,F] ; CLOSED =[B,A] ; X =[B]
4. OPEN=[C,F] ; CLOSED =[G,B,A] ; X =[G]
5. OPEN=[F,E] ; CLOSED =[C,G,B,A] ; X =[C]
6. OPEN=[E,J] ; CLOSED =[F,C,G,B,A] ; X =[F]
7. OPEN=[J,D,I] ; CLOSED =[E,F,C,G,B,A] ; X =[E]
8. OPEN=[D,I] ; CLOSED =[J,E,F,C,G,B,A] ; X =[J]
9. OPEN=[I] ; CLOSED =[D,J,E,F,C,G,B,A] ; X =[D]
10. OPEN=[H,K] ; CLOSED =[I,D,J,E,F,C,G,B,A] ; X =[I]
11. OPEN=[K] ; CLOSED =[H,I,D,J,E,F,C,G,B,A] ; X =[H]
12. OPEN=[] ; CLOSED =[H,I,D,J,E,F,C,G,B,A]; X =[K] **“K” WAS FOUND.**

LIST : A,B,G,C,F,E,J,D,I,H,”K”

* 1. List the nodes in the open and close lists for every iteration while searching for *J* from *B*, using best first search. Use the costs in the links as g(n), and assume that thedistance to the goal is the minimum skip distance; that is, h(n) is the minimum number of links between a node and the goal.

|  |  |  |
| --- | --- | --- |
| X | OPEN | CLOSED |
|  | B3 |  |
| B3 | C4,F4 | B3 |
| C4 | E3,F4 | B3,C4 |
| E3 | D1,I2,F4 | B3,C4,E3 |
| D1 | I2,F4 | B3,C4,E3,D1 |
| I2 | H1,J1,K1,F4 | B3,C4,E3,D1,I2 |
| H1 | J1,K1,F4 | B3,C4,E3,D1,I2,H1 |
| **J1** | K1,F4 | B3,C4,E3,D1,I2,H1 |

**“J” WAS FOUND**

1. What changes would you make to the BEST – FIRST algorithm, to convert it to a Hill Climbing algorithm, Write the algorithm and explain the changes.

From what we discussed, as a group we agreed that to obtain the Hill Climbing algorithm we should remove the open and close lists from the Best - First algorithm, because Hill Climbing algorithms are based only on the consequences of their immediate actions and they don’t need to save a history of the visited nodes. In Hill Climbing algorithm the search ends if the next node to move is worse than the actual node.

Discrete Space Hill Climbing Algorithm;

begin  
 currentNode = startNode;  
 loop do  
 L = NEIGHBORS(currentNode);  
 nextEval = -INF;  
 nextNode = NULL;  
 for all x in L   
 if (EVAL(x) > nextEval)  
 nextNode = x;  
 nextEval = EVAL(x);  
 if nextEval <= EVAL(currentNode)  
 //Return current node since no better neighbors exist  
 return currentNode;  
 currentNode = nextNode;