CS 540-1: Introduction to Artificial Intelligence
Homework Assignment # 2

Assigned: 9/13
Due: 9/20 before class

Hand in your homework:

This homework includes one written question and one programming question. Please type the written portion and hand in a pdf file named **hw2.pdf**. For the programming question, hand in the Java program **successor.java**. Go to UW Canvas, choose your CS540-1 course, choose Assignment, click on Homework 2: this is where you submit your files.

Problem 1. Search [60]

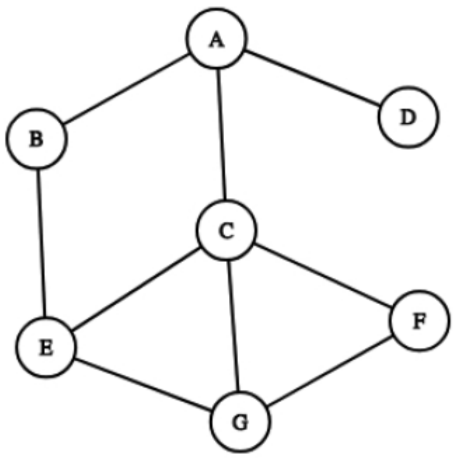

Figure 1: The unweighted, undirected graph

Given the undirected, unweighted graph in Figure 1, find the shortest path **from A to G** using different search algorithms.

You are to "run" the algorithms by hand. Use a CLOSED data structure to avoid cycles. Specifically, for uninformed search and uniform cost search follow slides 28 and 43 (Hint: these slides allow multiple copies of the same node in OPEN); for A* search follow slide 21.

To break ties, assume nodes are expanded (taken out of the OPEN data structure) in alphabetical order with everything else being equal.

For each algorithm, write down :

- States of OPEN and CLOSED in each step of the algorithm, together with the appropriate back pointer (see example steps below)
- The solution path found

a) [10] Breadth First Search.

The first few iterations of Breadth First Search have been shown to help you start.

OPEN (Queue)[Top is towards right]	**CLOSED**
A(Null)	
D(A), C(A), B(A)	A(Null)
E(B), A(B), D(A), C(A)	A(Null), B(A)
...	...

Solution:

- "Discarding nodes if they are already present in CLOSED", these steps are present with no action/changes in CLOSED column but with node removed in OPEN.

OPEN	**CLOSED**
$A_1(Null)$	
$B_2(A), C_3(A), D_4(A)$	$A_1(Null)$
$C_3(A), D_4(A), A_5(B), E_6(B)$	$A_1(Null), B_2(A)$
$D_4(A), A_5(B), E_6(B), A_7(C), E_8(C), F_9(C), G_10(C)$	$A_1(Null), B_2(A), C_3(A)$
$A_5(B), E_6(B), A_7(C), E_8(C), F_9(C), G_{10}(C)$	$A_1(Null), B_2(A), C_3(A), D_4(A)$
$E_6(B), A_7(C), E_8(C), F_9(C), G_{10}(C)$	$A_1(Null), B_2(A), C_3(A), D_4(A)$
$A_7(C), E_8(C), F_9(C), G_{10}(C), B_{11}(E), C_{12}(E), G_13(E)$	$A_1(Null), B_2(A), C_3(A), D_4(A), E_6(B)$
$E_8(C), F_9(C), G_{10}(C), B_{11}(E), C_{12}(E), G_{13}(E)$	$A_1(Null), B_2(A), C_3(A), D_4(A), E_6(B)$
$F_9(C), G_{10}(C), B_{11}(E), C_{12}(E), G_{13}$	$A_1(Null), B_2(A), C_3(A), D_4(A), E_6(B)$
$G_{10}(C), B_{11}(E), C_{12}(E), G_{13}(E)$	$A_1(Null), B_2(A), C_3(A), D_4(A), E_6(B), F_9(C)$

The BFS is thus given by, $A(NULL) -> C(A) -> G(C)$

b) [10] Depth First Search.

Solution:

- "Discarding nodes if they are already present in CLOSED", these steps are present with no action/changes in CLOSED column but with node removed in OPEN.

OPEN	**CLOSED**
$A_{-1}(NULL)$	
$B_{-4}(A), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL)$
$A_{-6}(B), E_{-5}(B), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A)$
$E_{-5}(B), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A)$
$B_{-9}(E), C_{-8}(E), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B)$
$C_{-8}(E), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B)$
$A_{-13}(C), E_{-12}(C), F_{-11}(C), G_{-10}(C), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B), C_{-8}(E)$
$E_{-12}(C), F_{-11}(C), G_{-10}(C), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B), C_{-8}(E)$
$F_{-11}(C), G_{-10}(C), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B), C_{-8}(E)$
$G_{-10}(C), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B), C_{-8}(E), F_{-11}(C)$

The DFS is thus given by, $A(NULL) -> B(A) -> E(B) -> C(E) -> F(C) -> G(F)$

c) [10] Iterative Deepening. Let us use the following convention: for the first depth cutoff (the very first iteration of the outer loop) we will goal check A and also A's successors, but nothing further.

Solution:

- "Discarding nodes if they are already present in CLOSED", these steps are present with no action/changes in CLOSED column but with node removed in OPEN.

Iteration 1:

OPEN	**CLOSED**
$A_{-1}(Null)$	
$B_{-4}(A), C_{-3}(A), D_{-2}(A)$	$A_{-1}(Null)$
$C_{-3}(A), D_{-2}(A), A_{-5}(B)$	$A_{-1}(Null), B_{-4}(A)$
$D_{-2}(A), A_{-5}(B), A_{-6}(C)$	$A_{-1}(Null), B_{-4}(A), C_{-3}(A)$
$A_{-5}(B), A_{-6}(C), A_{-7}(D)$	$A_{-1}(Null), B_{-4}(A), C_{-3}(A), D_{-2}(A)$
$A_{-6}(C), A_{-7}(D)$	$A_{-1}(Null), B_{-4}(A), C_{-3}(A), D_{-2}(A)$
$A_{-7}(D)$	$A_{-1}(Null), B_{-4}(A), C_{-3}(A), D_{-2}(A)$

Iteration 2:

OPEN	**CLOSED**
$A_{-1}(NULL)$	
$B_{-4}(A), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL)$
$A_{-6}(B), E_{-5}(B), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A)$
$E_{-5}(B), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A)$
$B_{-9}(E), C_{-8}(E), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B)$
$C_{-8}(E), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B)$
$A_{-13}(C), E_{-12}(C), F_{-11}(C), G_{-10}(C), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B), C_{-8}(E)$
$E_{-12}(C), F_{-11}(C), G_{-10}(C), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B), C_{-8}(E)$
$F_{-11}(C), G_{-10}(C), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B), C_{-8}(E)$
$G_{-10}(C), G_{-7}(E), C_{-3}(A), D_{-2}(A)$	$A_{-1}(NULL), B_{-4}(A), E_{-5}(B), C_{-8}(E), F_{-11}(C)$

The IDS is thus given by, $A(NULL) -> B(A) -> E(B) -> C(E) -> F(C) -> G(F)$

Now suppose we have different edge costs and heuristic function h as in Figure 2.

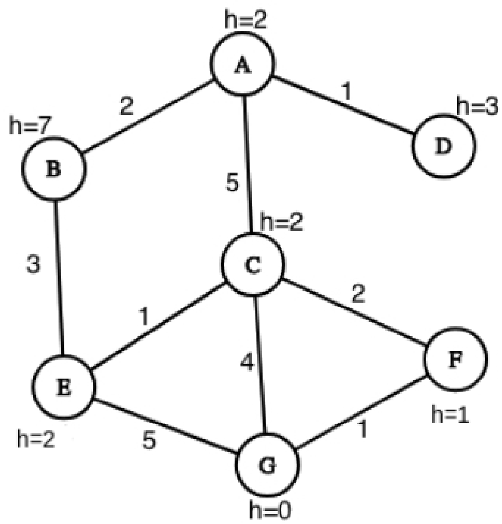

Figure 2: The weighted graph and the $h(\cdot)$ function

For parts (d) and (e) also include the node score (for A* this is $g + h$) in your answer alongside with the back pointer.

d) [10] Perform **Uniform Cost Search**. The first few steps are shown for Uniform Cost Search in the example table below.

OPEN (Priority Queue)[Top is towards right]	**CLOSED**
A(0,Null)	
C(5,A), B(2,A), D(1,A)	A(0,Null)
...	...

Solution:

- "Discarding nodes if they are already present in CLOSED", these steps are present with no action/changes in CLOSED column but with node removed in OPEN.
- Replacing nodes with less cost against those with greater

OPEN	**CLOSED**
$A_0(NULL)$	
$B_2(A), C_5(A), D_1(A)$	$A_0(NULL)$
$B_2(A), C_5(A)$	$A_0(NULL), D_1(A)$
$A_4(B), E_5(B), C_5(A)$	$A_0(NULL), D_1(A), B_2(A)$
$E_5(B), C_5(A)$	$A_0(NULL), D_1(A), B_2(A)$
$A_10(C), E_6(C), F_7(C), G_9(C), E_5(B)$	$A_0(NULL), D_1(A), B_2(A), C_5(A)$
$A_{10}(C), E_6(C), F_7(C), G_9(C), B_8(E), C_6(E), G_{10}(E)$	$A_0(NULL), D_1(A), B_2(A), C_5(A), E_5(B)$
$A_{10}(C), E_6(C), F_7(C), G_9(C), B_8(E), G_{10}(E)$	$A_0(NULL), D_1(A), B_2(A), C_5(A), E_5(B)$
$C_9(F), G_8(F), A_{10}(C), E_6(C), G_9(C), B_8(E), G_{10}(E)$	$A_0(NULL), D_1(A), B_2(A), C_5(A), E_5(B), F_7(C)$
$C_9(F), A_{10}(C), E_6(C), G_9(C), B_8(E), G_{10}(E)$	$A_0(NULL), D_1(A), B_2(A), C_5(A), E_5(B), F_7(C)$

The UCS with shortest path from A to G is thus given by, $A_0(NULL) -> C_5(A) -> F_7(C) -> G_8(F)$

e) [20] Perform **A* search** with the given heuristic function $h(\cdot)$ in Figure 2.

Solution:

- "Discarding nodes if they are already present in CLOSED", these steps are present with no action/changes in CLOSED column but with node removed in OPEN.
- Replacing nodes with less cost against those with greater

OPEN	**CLOSED**
$A_{0+2}(NULL)$	
$B_{2+7}(A), C_{5+2}(A), D_{1+3}(A)$	$A_{2=0+2}(NULL)$
$B_{2+7}(A), C_{5+2}(A)$	$A_{2=0+2}(NULL), D_{4=1+3}(A)$
$B_{2+7}(A), A_{10+2}(C), E_{6+2}(C), G_{9+0}(C), F_{7+1}(C)$	$A_{2=0+2}(NULL), D_{4=1+3}(A), C_{7=5+2}(A)$
$B_{2+7}(A), A_{10+2}(C), G_{9+0}(C), F_{7+1}(C), C_{7+2}(E)$	$A_{2=0+2}(NULL), D_{4=1+3}(A), C_{7=5+2}(A), E_{8=6+2}(C)$
$B_{2+7}(A), A_{10+2}(C), G_{8+0}(F), C_{7+2}(E)$	$A_{2=0+2}(NULL), D_{4=1+3}(A), C_{7=5+2}(A), E_{8=6+2}(C), F_{8=7+1}(C)$
$B_{2+7}(A), A_{10+2}(C), C_{7+2}(E)$	$A_{2=0+2}(NULL), D_{4=1+3}(A), C_{7=5+2}(A), E_{8=6+2}(C), F_{8=7+1}(C), G_{8=}$

The A* Search from A to G is given by, $A_2(NULL) -> C_7(A) -> F_8(C) -> G_8(F)$