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

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

Problem 1. Search [60]

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

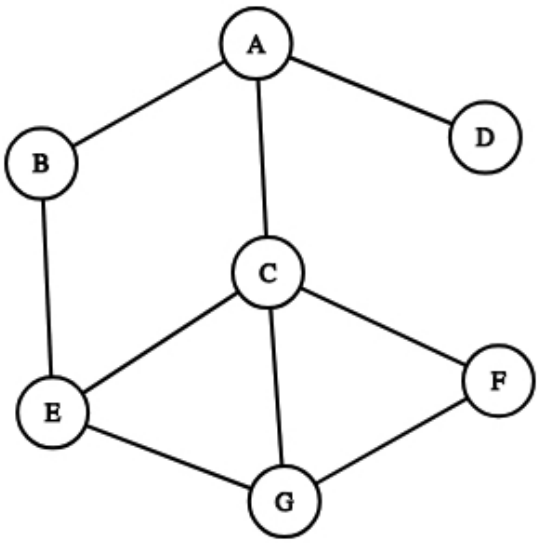

Figure 1: The unweighted, undirected graph

a) [10] Breadth First Search.

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)
G(C),F(C),E(C),A(C),E(B),A(B),D(A)	A(Null),B(A),C(A)
A(D),G(C),F(C),E(C),A(C),E(B),A(B)	A(Null),B(A),C(A),D(A)
Reasoning: Ignore A(B) as A $\in$ CLOSED	
A(D),G(C),F(C),E(C),A(C),E(B)	A(Null),B(A),C(A),D(A)
G(E),C(E),B(E),A(D),G(C),F(C),E(C),A(C)	A(Null),B(A),C(A),D(A),E(B)
Reasoning: Ignore A(C),E(C) as A,E $\in$ CLOSED	
G(E),C(E),B(E),A(D),G(C),F(C)	A(Null),B(A),C(A),D(A),E(B)
G(F),C(F),G(E),C(E),B(E),A(D),G(C)	A(Null),B(A),C(A),D(A),E(B),F(C)

Goal check G(C) and then use back pointers in CLOSED : $G \rightarrow C \rightarrow A$ to get the path Solution Path is : $A \rightarrow C \rightarrow G$

b) [10] Depth First Search.

OPEN (Stack) [Top is towards right]	**CLOSED**
A(Null)	
D(A), C(A), B(A)	A(Null)
D(A), C(A), E(B), A(B)	A(Null), B(A)
Reasoning: Ignore A(B) as A $\in$ CLOSED	
D(A), C(A), G(E), C(E), B(E)	A(Null), B(A), E(B)
Reasoning: Ignore B(E) as B $\in$ CLOSED	
D(A), C(A), G(E), G(C), F(C), E(C), A(C)	A(Null), B(A), E(B), C(E)
Reasoning: Ignore A(C),E(C) as A,E $\in$ CLOSED	
D(A), C(A), G(E), G(C), G(F), C(F)	A(Null), B(A), E(B), C(E), F(C)
Reasoning: Ignore C(F) as C $\in$ CLOSED	
D(A), C(A), G(E), G(C), G(F)	A(Null), B(A), E(B), C(E), F(C)

Goal check G(F) and use backpointers in CLOSED : $G \rightarrow F \rightarrow C \rightarrow E \rightarrow B \rightarrow A$ to get path Solution: $A \rightarrow B \rightarrow E \rightarrow C \rightarrow F \rightarrow G$

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.

OPEN (Stack)[Top is towards right]	**CLOSED**
Iteration 1	
A(Null)	
D(A), C(A), B(A)	A(Null)
D(A), C(A)	A(Null), B(A)
D(A)	A(Null), B(A), C(A)
	A(Null), B(A), C(A), D(A)
Iteration 2	
A(Null)	
D(A), C(A), B(A)	A(Null)
D(A), C(A), E(B), A(B)	A(Null), B(A)
Reasoning: Ignore A(B) as A $\in$ CLOSED	
D(A), C(A)	A(Null), B(A), E(B)
D(A), G(C), F(C), E(C), A(C)	A(Null), B(A), E(B), C(A)
Reasoning: Ignore A(C), E(C) as A,E $\in$ CLOSED	
D(A), G(C), F(C)	A(Null), B(A), E(B), C(A)
D(A), G(C)	A(Null), B(A), E(B), C(A), F(C)

Goal check G(C) and then use backpointers in the CLOSED : $G \rightarrow C \rightarrow A$ to get the path
Solution Path is : $A \rightarrow C \rightarrow G$

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

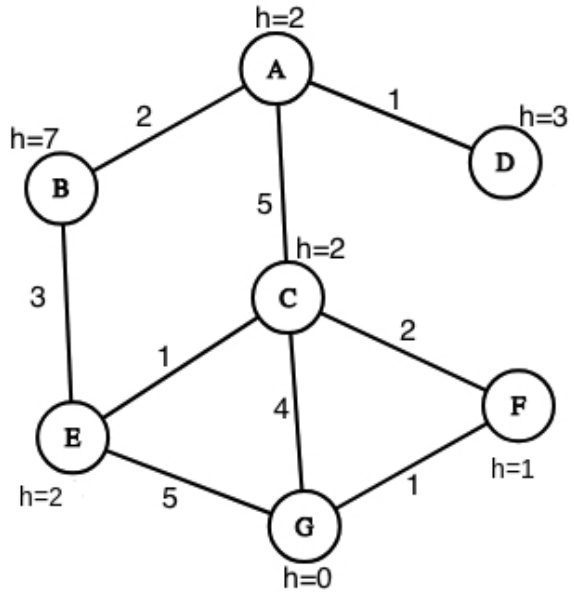

Figure 2: The weighted graph and the $h(\cdot)$ function

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**.

OPEN (Priority Queue)[Top towards right]	**CLOSED**
A(0,Null)	
C(5,A), B(2,A), D(1,A)	A(0,Null)
C(5,A), B(2,A), A(2, D)	A(0,Null), D(1,A)
Reasoning: Ignore A(2,D) as A ∈ CLOSED	
C(5,A), B(2,A)	A(0,Null), D(1,A)
E(5,B), C(5,A), A(4,B)	A(0,Null), D(1,A), B(2,A)
Reasoning: Ignore A(4,B) as A ∈ CLOSED	
E(5,B), C(5,A)	A(0,Null), D(1,A), B(2,A)
A(10,C), G(9,C), F(7,C), E(6,C), E(5,B)	A(0,Null), D(1,A), B(2,A), C(5,A)
G(10,E),A(10,C),G(9,C),B(8,E),F(7,C),E(6,C),C(6,E)	A(0,Null),D(1,A),B(2,A),C(5,A),E(5,B)
Reasoning: Ignore E(6,C),C(6,E) as E,C ∈ CLOSED	
G(10,E),A(10,C),G(9,C),B(8,E),F(7,C)	A(0,Null),D(1,A),B(2,A),C(5,A),E(5,B)
G(10,E),A(10,C),G(9,C),C(9,F),G(8,F),B(8,E)	A(0,Null),D(1,A),B(2,A),C(5,A),E(5,B),F(7,C)
Reasoning: Ignore B(8,E) as B ∈ CLOSED	
G(10,E),A(10,C),G(9,C),C(9,F),G(8,F)	A(0,Null),D(1,A),B(2,A),C(5,A),E(5,B),F(7,C)

Goal check G(8,F) and then use backpointers in the CLOSED : $G \to F \to C \to A$ to get the path
Solution Path is : $A \to C \to F \to G$

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

OPEN (Priority Queue)	**CLOSED**
A_{0+2}(Null)	
B_{2+7}(A), C_{5+2}(A), D_{1+3}(A)	A_{0+2}(Null)
B_{2+7}(A), C_{5+2}(A)	A_{0+2}(Null), D_{1+3}(A)
Reasoning: A_{2+2}(D) is generated as a successor, but its $g(A) = 2$ is larger than the $g(A)$ of the old A_{0+2}(Null) in CLOSED. By slide 21, it is ignored.	
G_{9+0}(C), B_{2+7}(A), F_{7+1}(C), E_{6+2}(C)	A_{0+2}(Null), D_{1+3}(A), C_{5+2}(A)
Reasoning: A_{10+2}(C) is ignored because its $g(A) = 10 > 0$	
G_{9+0}(C), B_{2+7}(A), F_{7+1}(C)	A_{0+2}(Null), D_{1+3}(A), C_{5+2}(A), E_{6+2}(C)
B_{9+7}(E) $g(B) = 9 > 2$ C_{7+2}(E) are ignored because $g(C) = 7 > 5$ G_{11+0}(E) $g(G) = 11 > 9$	
B_{2+7}(A), G_{8+0}(F)	A_{0+2}(Null), D_{1+3}(A), C_{5+2}(A), E_{6+2}(C), F_{7+1}(C)
Reasoning: C_{9+2}(F) is ignored because its $g(C) = 9 > 5$	
B_{2+7}(A)	A_{0+2}(Null), D_{1+3}(A), C_{5+2}(A), E_{6+2}(C), F_{7+1}(C), G_{8+0}(F)

Goal check G_{8+0}(F) and then use backpointers in the CLOSED : $G \to F \to C \to A$ to get the path
Solution Path is : $A \to C \to F \to G$

NOTE: The solution paths of UCS and A* are the same as the heuristic $h(.)$ used here is admissible