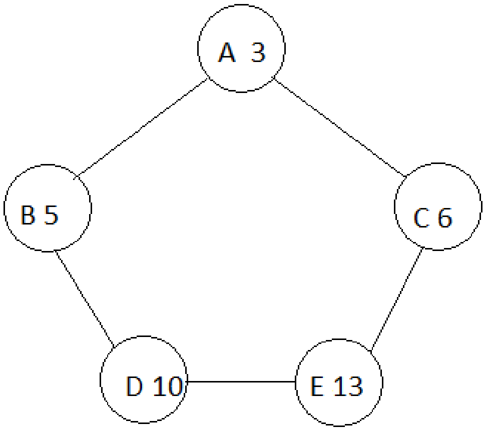
Problem Set 1

Assigned: January 23
Due: February 1

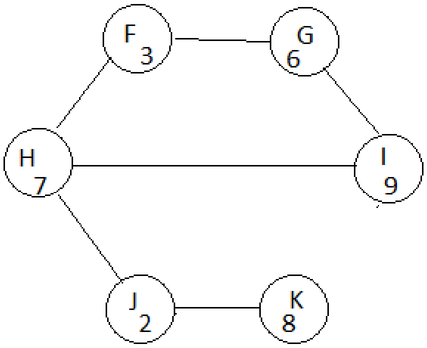
PLEASE NOTE: All assignments are due at the start of class, 11:00 AM, *NOT* at the end of the day.

The "Best Independent Set" problem is defined as follows:
Input: An undirected graph G, in which each vertex is marked by a positive value ; and a target value T.
Goal: A set of vertices S such that (a) no two vertices in S are connected by an edge in G; and (b) the total value of the vertices in S is at least T.

For instance, in Graph 1 below, with $T = 16$, there are three solutions: { C,D }, { A,E }, and { B,E } In Graph 2, with $T = 21$ the only solution is { G,H,K }



Graph 1



Graph 2

The N-Queens problem is actually a special case of this, with the vertices being the squares on the chess board, an edge between two vertices if the squares attack, all vertices have value 1, and the target value is N.

Problem 1

Consider the following state space for systematically solving the problem:

State: A set of vertices, no two of which are connected by an edge.
Successor to state S: Add some vertex V to S such that (a) V is alphabetically later than the vertices in S (to guarantee systematic search); (b) V is not connected to any vertex in S.

Start state: The empty set.
Goal state: A state that attains the target value.

Show the part of the state space that would be generated if you use depth-first search in this state space to find the solution to graph 2 with $T = 21$. You should show this as a tree. I recommend drawing trees for assignments left-to-right typewriter format like this:

```
Root--->C1--->C1.1--->C1.1.1
     |     |       |
     |     |       |->C1.1.2
     |     |
     |     |->C1.2--->C1.2.1
     |
     |->C2--->C2.1
```

If you want to use graphics software to make a nice looking tree, that's even better, though more work for you. If you want to draw it out and photograph it, that's OK, though less preferred.

Problem 2

Show the states generated in doing a breadth-first search of the state space in problem 1.

Problem 3

A. Describe an instance of Best Independent Set where doing a breadth-first search will construct 1000 (or more) times as many states as doing a depth-first search.

B. Describe an instance of Best Independent Set where doing a depth-first search will construct 1000 (or more) times as many states as doing a breadth-first search. (Hint: There is an instance where the correct solution contains only one vertex.)

Problem 4

Consider the Best Independent Set on a graph in general with n vertices (*not* on the specific examples above).

What is the maximal depth D of the state space? What is the branching factor B? Give a bound on the number of states in the state space (you should be able to get a bound much smaller than B^D.)

Problem 5

Consider trying to solve Best Independent Set using hill climbing in the following state space.

State: Any set of vertices.
Neighbors of state S: Either add one vertex to S or delete one vertex from S.
Error function:

Max(0,$T-$(Total value of the vertices in S)) +
sum of the cost of all edges that connect two vertices in S
where the cost of an edge is considered to be the value of its lower end.

For example, in graph 1, suppose S= { A,C } and $T = 16$.
Edge $A - C$ is in G, value(A)=3, value(C)=6, so cost(A-C)=3.
So Error(S) = Max(0,(16$-$9))+3 = 10.

What is the sequence of states encountered doing simple hill-climbing in this space, to solve this problem for Graph 2, starting from state {F,G,H,I,J } with a target of $T = 20$?