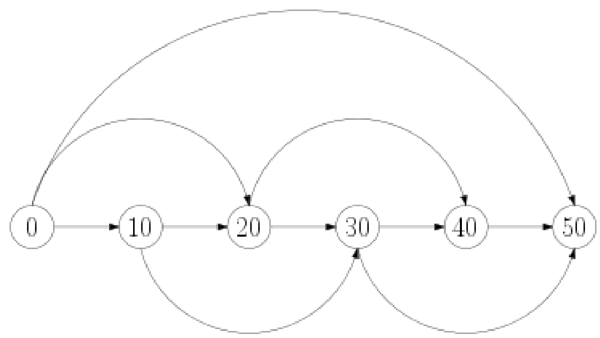
Due : December 13th, 2016

1. Imagine that you have an infinite amount of coins with values 10, 20, and 50. You can verify that the minimum amount of coins needed to make some amount, $x \in \{0, 10, 20, 30, 40, 50\}$, is equal to the shortest path between the node labeled 0 and the node labeled x in the digraph below (consider all edge costs as 1).



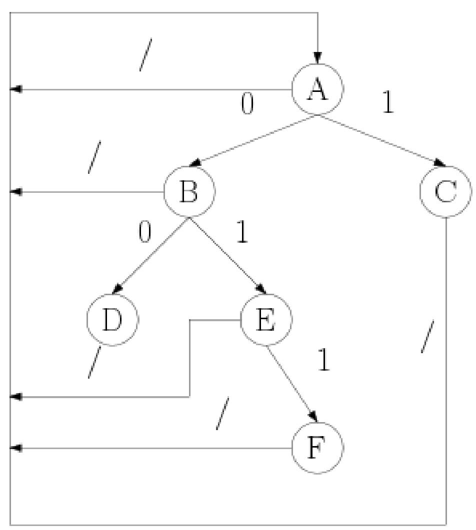
   (a) We modify the problem a little by adding more currency. Assume that now you have coins with values 10, 30, 50 and bills with value 100. Also, we will add a "weight" to the use of each type of currency. Draw a graph similar to the one above (but with the modified node values and edge weights) so that the shortest path from the node labeled 0 to the node labeled $x \in \{0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ denotes the minimum weight needed to reach the value x. The weight for each currency is given below:

Value	10	30	50	100
Weight	5	10	15	1

   (b) Write pseudocode that returns the minimum weight needed to reach the value x. Let C be the set of currencies, with $|C| = n$. The values and weights of the currency is given in arrays V and W, such that $V[c]$ and $W[c]$ is the value and weight of currency $c \in C$. Assume that initially you are given one node labeled 0. You may also use a method $createEdge(a, b, weight)$, which builds a directed edge from a to b with weight $weight$ (which also builds the node(s) if they do not exist). If you use more than two instances $createEdge(a, b, weight)$ with the same (ordered) pair (a, b). We will only consider the last method call.

(c) Can the graph you constructed in (b) be considered as an instance of a DAG (Directed Acyclic Graph)? If so, explain why it is. If not, find a counterexample.

2. For the text compression problem, assume that we use an explicit 'ending symbol' for the sole purpose of marking the end of each encoded letter. In this case, it doesn't matter if our code is prefix-free since we always know when an encoded letter terminates. For example, if the encoding/decoding is done in the manner illustrated below, the string 'BAFC' would be encoded to '0//011/1/', where '/' is our ending symbol.



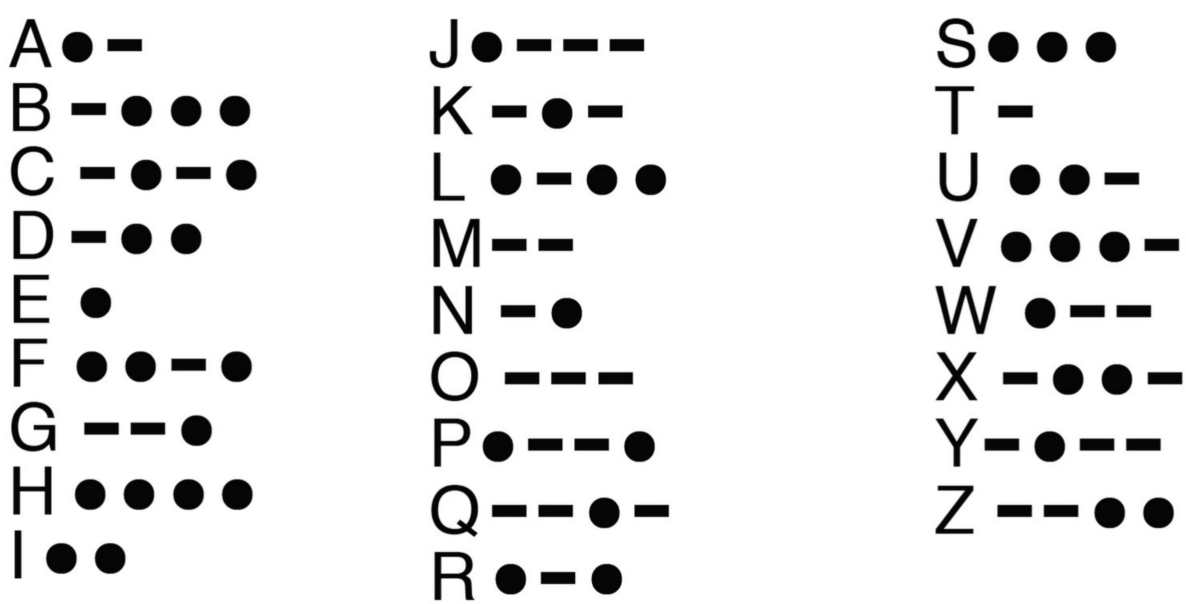
(a) Write the corresponding code for the string 'ABCDEF' and the total number of (node to node) transitions needed for decoding.

(b) Bob states that if the *relative frequency* of a letter (the percentage of times it appears in the string) is no less than the sum of the frequency of its descendants, then the total time needed for decoding the whole code would be optimal (that is, the number of transitions is minimal). Is Bob right or wrong? If Bob is right, justify his statement. If not (whether Bob's optimality condition is insufficient or totally wrong), give a counterexample. You are allowed to set the frequency for each of the six letters and use the compression described above, but you must clearly state that you used the above encoding method. Otherwise, you may design a new compression scheme (using 0, 1 and one 'ending symbol' '/') and assign a frequency for each character.

(c) Encode the following letters in a way so that the running time for the conversion would be optimal (you may only use 0, 1 and one 'ending symbol' '/'). Draw a graph similar to above.

Letter	a	b	c	d	e
Frequency	0.4	0.25	0.15	0.1	0.1

3. Morse Code is a common encoding scheme used by radio operators. Morse Code uses a series of *dits* (. or *dot*) and *dahs* (- or *dash*) in order to encode text, similar to how we use 0 and 1. Below is a Morse Encoding chart. Determine whether or not it is a prefix-free code. If it is, give a valid decoding tree for Morse Code. If it is not, give an example of a **valid** English word (of length greater than 5) that has an ambiguous encoding. In this case, clearly state the word, its Morse Encoding, and another string that has the same encoding.



4. Consider the following problems about weighted, undirected graphs. If you need to draw graphs, it is acceptable to draw them on paper and include a photo with your submission (in the same file, not a separate one). Be sure to draw **very** neatly. Your examples should all have at least 5 vertices and 6 edges.

   (a) Give a graph that has exactly one minimum spanning tree. Give a graph that has exactly two minimum spanning trees.

   (b) Give a graph where the edge with the most weight is in all minimum spanning trees of the graph.

   (c) Give a graph that has at least two minimum spanning trees **and** some pair of them have no edges in common. Show the pair of minimum spanning trees as well.