ECE 3484 - Combinatorial Optimization - Homework #1

Solve the following problems. Proofs should be as precise and concise as possible. Clearly state any results that your arguments depend on. All graphs are undirected unless otherwise specified. You are encouraged to discuss the problems with other students, but you should write the solutions on your own.

1. Consider the following implementation of a bubble sort. Given an array a[] containing n values, the algorithm repeatedly scans the list from top to bottom. In each pass, adjacent entries are compared and swapped if they are in the wrong order.

```
for (i=0; i<n-1; i++) {
  for (j=0; j<n-1-i; j++)
    if (a[j+1] < a[j]) {  /* compare the two neighbors */
      tmp = a[j];         /* swap a[j] and a[j+1]     */
      a[j] = a[j+1];
      a[j+1] = tmp;
  }
}
```

   Characterize the running time of this algorithm, and the best-case and worst-case running times, using asymptotic notation. Give *all* asymptotic bounds on these values that exist using O, Ω, and Θ, assuming each instruction takes a constant amount of time.

2. Consider a sequential search algorithm. Given a value, the algorithm scans a linear array until the value is found. If the value is not found, that algorithm returns a fail cod.e

   Characterize the best-case and worst-case running times of this algorithm using asymptotic notation. Give *all* asymptotic bounds on these values that exist using O, Ω, and Θ, assuming each instruction takes a constant amount of time.

3. Prove the following, or give a counter example:

   (a) $f(n) = \Omega(g(n))$ and $g(n) = \Omega(h(n))$ implies $f(n) = \Omega(h(n))$.

   (b) $f(n) = O(g(n))$ implies $g(n) = O(f(n))$.

   (c) $f(n) + g(n) = \Theta(min(f(n), g(n)))$.

4. Give optimization and decision versions of the following problems:

   (a) Minimum Spanning Tree

   (b) Maximum Matching

5. Argue that each of the following problems is or is not in NP (this is a hard problem):

   (a) Given a weighted graph $G = (V, E)$ and an integer k, is there at least one tour with cost $\leq k$?

   (b) Given a weighted graph $G = (V, E)$ and an integer k, is the number of tours with cost $\leq k$ equal to one million?

(c) Given a weighted graph $G = (V, E)$ and an integer k, is the number of tours with cost $\leq k$ greater than or equal to one million?

(d) Given a weighted graph $G = (V, E)$ and an integer k, does the one millionth-shortest tour have cost $\leq k$? (a tour t is the *n-shortest tour* if the number of tours whose cost is less than the cost of t is $n - 1$. Assume that no two different tours have the same cost.)

(e) Given a weighted graph $G = (V, E)$ and an integer k, does the one millionth-shortest tour have cost $> k$? Continue to assume that no two different tours have the same cost.