Caveat

You have three hours to complete the test. I most likely will be unavailable via email when you take test. Any appropriate question I do answer will be posted on the class website. The test is open book and class notes and closed neighbor. Computer access is limited to the class website. Every problem has a straightforward answer. Not all questions have the same point value.

Page 1	**____ / 15**
Page 2	**____ / 15**
Page 3	**____ / 15**
Page 4	**____ / 20**
Page 5	**____ / 20**
Page 6	**____ / 15**
Total	**____ /100**

1. (5 points) The decision sorting problem determines whether its input is arranged in sorted order. Under what conditions is the decision sorting problem NP-hard (e.g., NP is a proper subset of P (not the right answer).

2. (10 points) Suppose a tour for a Travelling Salesperson instance equals [1, 9, 3, 4, 7, 8, 5, 6, 2] (i.e., *tour*[0] = 1, *tour*[1] = 9, *tour*[2] = 3, …). What is the tour after *interchange*(2, 7) is executed, where the *interchange*() method is the one described in assignment 8.

tour = [_____]

3. (15 points) The decision travelling salesperson problem takes as input a weighted graph and an amount *k*. The question under consideration is whether there is a tour whose length is at most *k*. Provide a *simple linear nondeterministic* algorithm that solves the problem, where the graph is represented using a two-dimensional array *w* (i.e., element *w*[*a*][*b*] is the length of edge (*a, b*)).

```
DTSPSolver( int w[1..n][1..n], int k ) {

   int[] t = new int array with n elements
   // Attempt to assign t so that it represents a tour with the desired
   // characteristics (i.e., a tour whose length ≤ k. Depending on w it
   // may or may not be successful
```

4. (15 points) Complete the following reduction for the CNF satisfiability problem for a formula *F*. The reduction is to the Halting problem. The reduction begins as follows.
   - Define algorithm p() in the following manner.

```
p(CNF F) {
   // perform exhaustive search on assigning values to the variables
   // in F (there are at most 2^n different assignments)
   Boolean found = false;
   for each possible assignment of 0 and 1 to the variables in F do
      a = the current assignment of 0s and 1s to the variables in F
      found = F(a); // F(a) is the value of F when its variables
                    // have the values in a
      if ( found ) {
         // found an assignment that makes F satisfiable
         break;
      }
   }
   // loop forever iff a satisfiable assignment was found
   while ( found ) {
       continue;
   }

}
```

   - Show that p() has infinite loop if and only if *F* is satisfiable.

5. (10 points) What is the importance of the Covey diagram for time management?

	Urgent	Non-urgent
Important		
Non-important		

6. (10 points) Explain why simulated annealing uses a Boltzmann-like criterion $e^{-delta/t}$ for determining whether to accept a mutation that worsens the solution. Your answer should take into account the roles of *delta* (the amount by which the solution is worsened) and the current temperature *t*.

7. (10 points) A mutation method allows simulated annealing to consider different solutions in its search for an optimal answer. For what follows, the problem of interest is 0/1 Knapsack.
   - Suppose the mutation method operates by replacing one item with another. When starting with a random solution is there always a way to get to an optimal solution? Explain.

   - Suppose the mutation method operates by either removing or adding one item. When starting with a random solution is there always a way to get to an optimal solution? Explain.

8. (10 points) A typical simulated annealing starting temperature is $2s$, where s^2 is an estimate of the variance of the scores in the solution space. Explain why $2s$ is a reasonable starting temperature for accepting most mutations given a Boltzmann-like criterion. (FYI, $e^{-1/2} \approx 0.606$ and $e^{-1} \approx 0.367$).

9. (15 points) A cake cutting is *fair* if each person believes that her or his slice equals at least $1/n^{th}$ of the cake, where n is the number of people. For the following subquestions assume: the people may have different perceptions of what is fair; each slicer attempts to produce a $1/n^{th}$ slice of the cake; and when a person chooses a slice of cake for her or his self, the person chooses a fair slice or better (if there is such a slice). Suppose there are three cake eaters X, Y, and Z.
    - Suppose X cuts the cake and assigns the slices to everyone. Does this cake cutting strategy necessarily produce a fair result? Explain why or why not.

    - Suppose X slices first and then Y. Suppose Z assigns the slices to everyone. Does this cake cutting strategy necessarily produce a fair result? Explain why or why not.

    - Suppose X slices first and then Y. Suppose Z assigns the slices to everyone in a manner that Y gets one of the two slices that Y produced. Does this cake cutting strategy necessarily produce a fair result? Explain why or why not.