Due date: February 20 @9:00AM

Submission Instructions:

- Each group must submit one solution to this assignment in class on the due date.
- Assignments must be neatly written or typed. If the grader cannot (easily) read an answer, she will give it a 0.
- Each group must submit only one solution. If multiple solutions are handed in, the grader will randomly discard all but one of them.
- Your submission must have the name of every group member clearly marked. If your name is not on it, you do not get credit.
- All arguments should be clear, concise, and (of course) correct. You will lose points for poor writing, bad grammar, etc...
- Use of the Internet for solving these problems is strictly forbidden, and will be treated as an honor code violation.
- Inter-group work: Members of different groups may discuss the questions. However:
  - With your submission you must provide a list of all non-group members with whom any member of your group discussed the assignment.
  - No written material may leave the inter-group discussion. If you talk with someone about the assignment, you must throw away and written notes at the end of the discussion.
Problem 1 (20 points): Recall from class the *classroom scheduling problem*: each class had a start and end time, and we wanted to assign the maximum possible number of non-overlapping classes to a single classroom.

Consider now the *weighted classroom scheduling problem*. Assume registration has already happened – we know the size of each class. So we want to assign not the largest number of classes to a classroom, but the most number of students. For example, we would prefer one class of 100 students over five classes of 10 students each.

- Formally state this problem, as we have done in class. That is, describe the input, the output, and the goal.
- Show that the algorithm used for solving the (unweighted) classroom scheduling problem will not work on the weighted classroom scheduling problem.
- Suppose we applied the same algorithm, but sorted by class size instead of finish time (take the largest class first, the next largest class that will fit, etc.) Prove or disprove that this algorithm always works.
Problem 2 (20 points): Tyrion Lannister is fleeing across the desert of Dorne, with Cersei’s knights in hot pursuit. He is following a fixed, pre-planned, straight-line path that passes several waterholes, and he needs to stop periodically to refill his water supply. He can carry enough water to make it $m$ miles; trying to go further than that is a death-wish. Waterholes are somewhat erratically placed, but are never more than $m$ miles apart. (In some case they might be considerably less.) However, given that Cersei’s men are catching up, and refilling water takes time, he wants to minimize the number of times he has to stop before he reaches safety in the city of Sunspear.

Given that the locations of all the waterholes along the path are known ahead of time, explain how he should pick which waterholes to stop to refill in order to minimize his total number of stops. Prove that your method will always return a solution that uses the minimum possible number of stops.
Problem 3 (20 points):
Use the definitions of $O()$, $\Omega()$, $\Theta()$ to prove each of the following:

1. $n^2 = O(n^3)$
2. $n = \Omega(n^{\frac{1}{2}})$
3. $\log_2 n = \Theta(\log_2 n^2)$
4. $n^{a+1} \neq O(n \log_b n)$ for any $a, b > 0$.

(Each should be proved from the definition: either a $N/c$ argument, or a limit argument. Do not just cite results from class for this problem – I want to see that you understand these.)