HW # 1

CS 401

Due Thu, Sept 10 at the beginning of class

Part I: Problems/Exercises From CLRS (3rd ed)

Ch 2: Exercise 2.3-3; Problem 2-3 (a and b only)
Ch 3: Exercises 3.1-2, 3.1-3, 3.1-4; Problem 3-4
Ch 4: Exercise 4.4-7

Part II: Code Analysis

Consider the code segments below:

```
// segment A
x = 0;
for(i=1; i<=n; i = i*2)
   for(j=1; j<n; j = j++)
      x++;


// segment B
x = 0;
for(i=1; i<=n; i=i*2)
   for(j=1; j<=i; j++)
      x++;
```

For each code segment give as tight a runtime analysis as you can as a function of n.

Part III: The Skyline Problem

You are to devise algorithm(s) for the following problem. Given a set of rectangular "buildings" all on the same horizon (i.e., no hills), you are to compute the resulting skyline.

A building is represented as a triple: (x_l, h, w) where x_l is the left boundary of the building, h is its height and w is its width (i.e, the right boundary is at $x_l + w$). The input to your algorithm is a collection of n such buildings. Note of course, that the buildings can overlap (e.g., one building can obscure part of another). You may assume that all coordinates are non-negative.

Without loss of generality, a skyline is represented by a sequence of horizontal line segments which give the outline of the skyline. A skyline starts at x-coordinate 0 and is a sequence of pairs (h, w) where h is a height and w is a width. The segments are assumed to be ordered left to right and each segment is assumed to begin where the previous ended (note that the segment might be at "street level").

Your job:

(A): Draw the skyline for $< (0,2),(2,4),(6,2),(0,3),(3,3) >$

(B): Suppose you are given a skyline and a new building. Describe an algorithm to "merge" the new building into the skyline.

(C): Using your answer from (B) give an algorithm for constructing a skyline from a given set of buildings which starts with a flat skyline and one-by-one merges each building into the skyline. Analyze the worst-case running time.

(D): Devise a faster divide and conquer algorithm for the problem. Hint: if you are given two skylines, how can you merge them into a single skyline and how long does it take? What is the worst case running time and why?

Part IV

You are given as set of n time intervals $(s_1, e_1), (s_2, e_2), ..., (s_n, e_n)$. The interval (s_i, e_i) indicates a start-time of s_i and an end time of e_i.

Think of each interval indicating that a "process" is active during that time.

We want to find a time at which a maximum number of processes are active.

Devise such an algorithm with running time $O(n \log n)$