Assignment 1

due Friday, January 20, 2006

1. Suppose that algorithm \( A \) uses \( \lceil 1000 \cdot n^2 \sqrt{n} \rceil \) operations while algorithm \( B \) uses \( 2 \cdot n^4 \) operations. Determine the value \( n_0 \) such that \( A \) is better than \( B \) for all \( n \geq n_0 \). [4 points]

2. exercise 3.1-4, p 50. Additionally, is \( 2^{2n+1} = O(2^{2n}) \)? [4 points]

3. exercise 3.2, p 58 [8 points]

4. exercise 3-3, part a (not part b), p 58. Ignore all functions involving \( \lg^* \) or a factorial (such as \( (\lg n)! \)). [8 points]

5. An algorithm takes 0.2\( \mu \) for input size 100 (this allows you to determine the constant, which will be different in each case). How much time does the algorithm take on an input of size 250 if the algorithm is . . . ?
   
   (a) \( \Theta(n) \)
   (b) \( \Theta(n \log n) \)
   (c) \( \Theta(n^3) \)
   (d) \( \Theta(2^n) \)

   [8 points]

6. Describe how to find the minimum and maximum of an array of \( n \) elements with at most \( \frac{3}{7} n \) element comparisons. (Do not count comparisons needed for the array indices.) [4 points]

Total: 36 points

Notes:

- A \( \mu \) is 1/1000 of a second.
- For Q5, let’s answer the question for a \( \Theta(n^2) \) algorithm. The algorithm uses time \( cn^2 \), for some \( c \). We can find \( c \): \( c(100)^2 = 0.2 \), or \( c = 0.2/10000 = 0.00002 \). The time for an input of size 250 is \( cn^2 = 0.00002(250)^2 = 1.25\mu \). Note that on an input 2.5 times as large it takes \((2.5)^2 = 6.25 \) times as long.
- Hint for Q6: form \( \lceil \frac{n}{2} \rceil \) pairs, from each pair find candidate min and candidate max for the whole list.