CMPU 241 - Analysis of Algorithms

Spring 2019

Practice Problem Set 1

You are encouraged to solve this problem set, but these problems will not be graded.

PS-1 Rank the following functions in terms of low (left) to high (right) asymptotic growth rates. Partition your list into equivalence classes such that functions $f(n)$ and $g(n)$ are in the same class (and surrounded by [ ]'s) if and only if $f(n) = \Theta(g(n))$. Use commas to separate different equivalence classes. If two or more functions have the same running time, group them with [ ]'s and show the complexity class that contains the functions.

For example, given the following sequence of functions: $n^{lg 2}$ n^3 n^{lg^4} n^{lg^8} n^2 , you should write the rankings as follows:

$$n^{lg 2} = \Theta(n),\ [n^2 = n^{lg 4} = \Theta(n^2)],\ [n^3 = n^{lg 8} = \Theta(n^3)]$$

Rank the following functions:

$$n^2 \quad n! \quad (\tfrac{3}{2})^n \quad n^3 \quad \ln n \quad 100 \quad 2^{lgn} \quad e^n \quad 4^{lgn} \quad n \quad 2^n \quad n\lg n \quad \lg^* n \quad n^{2.33} \quad 3^n \quad \lg n$$

PS-2 Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \le 2$. Justify your answers.

a. $T(n) = T(\frac{9n}{10}) + n$

b. $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$

c. $T(n) = T(n-1) + n$

d. $T(n) = 3T(\frac{n}{2}) + nlgn$

e. $T(n) = 6T(\frac{n}{4}) + n^2$

PS-3 (Problem 2.3-3) Use mathematical induction to show that when n is an exact power of 2, the solution of the recurrence

$$T(n) = \begin{cases} 2 & \text{if } n = 2 \\ 2T(\frac{n}{2}) + n & \text{if } n = 2^k, \text{ for } k > 1 \end{cases}$$

is $T(n) = nlgn$.

PS-4 Consider sorting n numbers stored in array A by first finding the smallest element of A and exchanging it with the number stored in $A[1]$. Then find the second smallest element of A, and exchange it with $A[2]$. Continue in this manner for the first $n-1$ elements of A. This algorithm, called SELECTION-SORT, is given below. Assume the input and output are as specified for the sorting problem on page 16 of our textbook.

```
Selection-Sort(A):
    n = A.length
    for j = 1 to n − 1
        smallest = j
        for i = j + 1 to n
            if A[i] < A[smallest]
                smallest = i
        exchange A[j] with A[smallest]
```

(a) Give a loop invariant for the Selection-Sort algorithm.

(b) Show that your algorithm is correct by giving a proof by induction (like that shown in class and given in Chapter 2 of our textbook) on the number of iterations of the outer for loop.