CSE100 Algorithm Design and Analysis — Homework set #1

Fall semester 2009 — Miguel Á. Carreira-Perpiñán

Total possible marks: 100. Homeworks must be solved individually. This set covers chapters 1–4 of the textbook *Introduction to Algorithms*, 3rd. ed., by Cormen et al.

Exercise 1: asymptotic behavior (20 points). Assume you have two computers, C_A and C_B capable of performing 10^6 and 10^8 operations per second, respectively. Both computers run a set of algorithms whose precise complexities $f(n)$ are given below. Determine the size n^* of the biggest input that can be processed in 1 second for each computer.

$f(n)$	n^* for C_A	n^* for C_B
$\lg \lg n$		
$\sqrt{n}$		
$14n + 4$		
$n \log_3 n + n$		
$n^2 + 7n$		
n^{10}		
2^n		
3^n		
$n!$		
n^n		

The precise complexity tells you how many operations are performed to solve an instance of size n. Assume each operation takes the same time and that the input sizes are natural numbers.

Exercise 2: running time (38 points). Consider the following function, where A is an array containing integers and v an integer.

```
FUNCTION-X(A, v)
s = 0
for i = 1 to A.length
    if A[i] == v
        s = s + 1
return s
```

Answer the following questions concisely:

1. (5 points) What does the function do?
2. (8 points) Give its running time $T(n)$ in asymptotic notation. What can you say about its worst, average and best case?
3. (10 points) Write the function using divide-and-conquer.
4. (10 points) Give the recurrence for $T(n)$ and solve it, thus giving its running time $T(n)$ in asymptotic notation.
5. (5 points) How much did $T(n)$ improve? Can you think of a way to reduce the running time order?

Exercise 3: recurrent equations (42 points). (This is based on textbook problems 4.1 and 4.3.) Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \le 2$.

1. (6 points) $T(n) = T(9n/10) + n\sqrt{n}$
2. (6 points) $T(n) = 16T(n/4) + n^2$
3. (6 points) $T(n) = T(\sqrt{n}) + 1$
4. (6 points) $T(n) = 7T(n/3) + n^2$
5. (6 points) $T(n) = 7T(n/2) + n^2 + 3n + 1$
6. (6 points) $T(n) = T(n-1) + n$
7. (6 points) $T(n) = 2T(n/4) + \sqrt{n}$

Justify your answers. If you use the master theorem, specify which case and show that its hypotheses are satisfied. If you use recursion trees to find a good guess, verify the guess with the substitution method.

Bonus exercise: recurrent equations (20 points). Consider the recurrence

$$T(n) = aT(n/b) + f(n).$$

Case 3 of the master theorem requires that $f(n) = \Omega(n^{\log_b a+\epsilon})$ for some constant $\epsilon > 0$ and that $f(n)$ satisfies the regularity condition $af(n/b) \le cf(n)$ for some constant $c < 1$ and all sufficiently large n. Prove that the regularity condition is always satisfied if $f(n) = n^k$. (This means we don't need to check it when f is a polynomial.)