Exercises Complexity Theory

Lecture 3

April 22, 2024

Only the exercises where points are given can be handed in.
(The maximum number of points per exercise is written in the margin.)

To be handed in on **April 29, 2024**, in Brightspace under Assignment 3, **deadline: 13:30.**

(40) **Exercise** 1. Suppose that for $k = 2, 4, 8, 10$, we have a recursive algorithm A_k which, on input n does $k(k+1) - 4$ recursive calls on an input of size $n/4$ and takes in addition $\Theta(n^{\frac{k}{2} - \frac{1}{3} \cdot (k \mod 4)})$ steps of work.
Let $T_k(n)$ denote the time complexity of A_k.
Determine functions g_k such that $T_k(n) = \Theta(g_k(n))$ for $k = 2, 4, 8, 10$.

Exercise 2. We have two recursive algorithms whose time complexities are, respectively, $T(n)$ and $S(n)$, which satisfy $T(n) = S(n) = n$ for $n = 1, 2, 3$ and for $n > 3$:

$$\begin{aligned} T(n) &= 3T(\frac{n}{2}) + n^2(2 + \cos n) \\ S(n) &= 3S(\frac{n}{2}) + 2n^2 \end{aligned}$$

(20) (a) Show that the Master Theorem cannot be applied to compute an f such that $T(n) = \Theta(f(n))$.

(10) (b) Give a g such that $S(n) = \Theta(g(n))$.

(20) (c) Use the g from (b) to make a guess for the f in (a) and prove that indeed $T(n) = \Theta(f(n))$.

(10) **Exercise** 3. We have a recursive algorithm whose time complexity $T(n)$ is given by:

$$T(n) = 16T(\frac{n}{4}) + n^2(n \mod 2)$$

Show that the Master Theorem cannot be applied to give an asymptotic bound for T.

Exercise 4. Prove that $T(n) = \Theta\left(n \log^2 n\right)$ for T with $T(n) = 2T(\frac{n}{2}) + n \log n$.

Exercise 5. Compute an f for which $T(n) = \Theta(f(n))$ for the following T

(a) $T(n) = 4T(\lceil n/3 \rceil) + n \log n$

(b) $T(n) = 4T(\lceil n/2 \rceil) + n^2\sqrt{n}$

(c) $T(n) = 3T(\lceil n/3 \rceil - 2) + n/2$

(d) $T(n) = T(\lfloor n/2 \rfloor) + T(\lfloor n/4 \rfloor) + T(\lfloor n/8 \rfloor) + n$

Exercise 6. (a) How would you adapt Karatsuba's algorithm for multiplication for two integers of n and m digits, where n, m are not powers of 2? Show that your algorithm is still $\Theta(n^{\log 3})$ for n the largest number of digits.

(a) How would you adapt Strassen's algorithm for multiplication for matrices of size $n \times n$, where n is not a power of 2? Show that your algorithm is still $\Theta(n^{\log 7})$.