2nd Homework Assignment

Spring Semester 2017
Prof. Helmut Alt

Introduction to Algorithms

Due March 20, 2017, 10 a.m.

Exercise 1

8 points

(a) Describe a simple algorithm to find the minimum and the maximum of a sequence of n elements (of a linearly ordered universe) and analyze the exact number of comparisons needed as a function of n.

(b) Solve the same problem with divide-and-conquer by partitioning sequences of length greater than 2 into two halves. Give an exact recursive equation for the number $C(n)$ of comparisons. You may assume that n is a power of 2.

Exercise 2

12 points

Consider the following Python function to compute the n-th Fibonacci number f_n:

```
def fib(n):
    if n == 0: return 0
    elif n == 1:
        return 1
    else:
        return fib(n-1) + fib(n-2)
```

(a) Show that this function has a runtime exponential in n. *Hint:* Count just the number $T(n)$ of additions. Set up a recursive equation for $T(n)$ and show by induction that it equals $f(n+1)-1$ for all n.

(b) What is, as a function of n in terms of Θ, the space needed by this algorithm?

(c) Find a function of linear runtime and constant space for the same problem.

*(d) Find a function of runtime $\Theta(\log n)$ for the same problem.
Hint: Observe that for $n \geq 2$,
$\begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix} = A \cdot \begin{pmatrix} f_{n-1} \\ f_{n-2} \end{pmatrix}$ where A is the matrix $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$

0 points for part (d), it's too difficult and just given as a challenge.