Institut for Matematik og Datalogi
Syddansk Universitet

September 22, 2015
JFB

Introduction to Computer Science
E15 – Discussion Sections (2) – Week 40

1. Page 244–245: Problem 5, 6, 7.

2. Pages 265–68: Problems 21, 40, 49, 50.

3. Design an algorithm for finding the square root of a positive integer, rounded down to the nearest integer. Thus, given input N, a positive integer, you should find a positive integer m such that $m^2 \leq N$, but $(m+1)^2 > N$. Use the binary search idea.

   (a) Express the algorithm in pseudocode.

   (b) Find a fundamental operation and use big theta notation to express how long your algorithm takes. Express this as a function of the positive integer N which is input.

4. Suppose you are given a list, P, of the first n primes, where n might be large. Design two algorithms that, when given an input, s, finds the smallest index, i, in the list $1 \leq i \leq n$, such that multiplying the ith and $i+1$st primes together gives a result at least as large as s. (Find i such that $P[i] * P[i+1] \geq s$.) The first algorithm should be based on sequential search and the second on binary search.

   (a) Write your algorithms in pseudocode. Explain why they work.

   (b) Analyze the running time of your algorithms (using Θ notation). You may assume that multiplying and comparing numbers takes 1 unit of time. (Think about why this assumption might not be reasonable.) Also, explain how your answer would change if you could assume that $s \leq n$.

5. Use pseudocode to write a recursive algorithm to compute the length of a list. Assume that you have a built-in function to get the next

entry in a list and another function to check if there are more items still left in the list. Analyze the running time of your algorithm (using Θ notation).