CS 161 Summer 1997

Handout #3

Problem Set 1 Due 3:05p, Tuesday, July 1, 1997

Readings

1. General course information: on-line, at

   `http://graphics.stanford.edu/cgi-bin/tolis/cs161-97-summer/getpage.cgi`

   These pages include *course registration*, information on the *proper format of your solutions*, as well as policies on *cooperation* and *extensions*.

2. Handouts 1, 2, 4, 5.
3. Chapters 1–3.
4. Sections 4.1–4.3, 6.1–6.4, 6.6, 8.1–8.2, 8.4.1

Exercises

Exercises are for you to work out, with the assistance of classmates and the course staff. Please don't hand in your solutions.

1. Exercise 1.3-5.
2. Problem 1-2.
3. Exercise 2.1-2.
4. Exercise 2.2-5.
5. Exercise 3.2-1.
6. Use both substitution and iteration to solve the recurrence

   $$T(n) = T(n-a) + T(a) + n,$$

   where $a \geq 1$ is constant.

7. Exercise 6.3-2.
8. Problem 8-4. In part (c) you should maintain the $\Theta(n \lg n)$ expected running time.
9. Describe a $\Theta(n \lg n)$ algorithm which, given a set S of n real numbers and another real number x, determines whether or not there exist two elements in S whose sum is exactly x.

Problems

Your problem set grade depends only on the solutions you hand in for the problems below.

1. **[1 point]** Fill out the on-line class registration form.

2. **[5 points]** Exercise 1.4-1.

3. **[3 points]** Exercise 1.3-6.

4. **[27 points]** Rank the following functions by order of growth, i.e. find an arrangement $g_1, g_2, \ldots, g_{25}$ of the functions such that each pair satisfies either $g_i = \Omega(g_{i+1})$ or $g_i = \Theta(g_{i+1})$ for $i = 1, 2, \ldots, 24$. Justify the relationship between each pair. (Unless otherwise stated, proofs will be required for all claims made on your homework assignments.) *Tip:* section 2.2 contains lots of useful goodies.

$$\begin{array}{ccccc}
(3/2)^n & (\sqrt{2})^{\lg n} & \lg^* n & n^2 & (\lg n)! \\
n^3 & \lg^2 n & \lg(n!) & 2^{2^n} & n^{1/\lg n} \\
\lg\lg n & n \cdot 2^n & n^{\lg\lg n} & \ln n & 2^n \\
2^{\lg n} & (\lg n)^{\lg n} & 4^{\lg n} & (n+1)! & \sqrt{\lg n} \\
n! & 2^{\sqrt{2\lg n}} & n & n\lg n & 1
\end{array}$$

5. **[5 points]** Evaluate the sum

$$\sum_{k=0}^{\lg n} k2^k$$

where n is an exact power of 2.

6. **[15 points]** Problem 4-4. You may assume that $T(n)$ is constant when $n \leq 8$ in part (b). Slopiness is acceptable *only* when it comes to floors and ceilings, i.e. you may assume that all arguments to $T(n)$, such as fractions as in $T(n/3)$, are integers.

7. **[15 points]** Problem 6-2.

8. **[3 points]** Exercise 8.2-1.

9. **[3 points]** Exercise 8.2-2.

10. **[3 points]** Exercise 8.2-4.

11. **[20 points]** Let u and v be two n bit numbers where for simplicity n is a power of 2. The traditional multiplication algorithm requires $O(n^2)$ operations. A divide-and-conquer based algorithm splits the numbers into two equal parts, computing the product as

$$uv = (a2^{n/2} + b)(c2^{n/2} + d) = ac2^n + (ad + bc)2^{n/2} + bd$$

The multiplications ac, ad, bc, and bd are done using the algorithm recursively. Determine this algorithm's running time. What is the running time if $ad + bc$ is computed as $(a + b)(c + d) - ac - bd$?