Introduction to Algorithms March 11, 2004
Massachusetts Institute of Technology 6.046J/18.410J
Professors Erik Demaine and Shafi Goldwasser Handout 12

Problem Set 4

This problem set is due **in recitation** on **Friday, March 19**.

Reading: Chapters §30.1-30.2, 11.1-11.3, 11.5, 12.1-12.3

There are **three** problems. Each problem is to be done on a **separate sheet** (or sheets) of paper. Mark the top of each sheet with your name, the course number, the problem number, your recitation section, the date, and the names of any students with whom you collaborated.

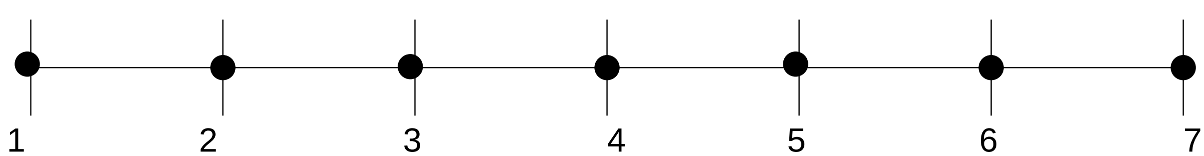
You will often be called upon to "give an algorithm" to solve a certain problem. Giving an algorithm entails:

1. A description of the algorithm in English and, if helpful, pseudocode.
2. A proof (or argument) of the correctness of the algorithm.
3. An analysis of the running time of the algorithm.

It is also suggested that you include at least one worked example or diagram to show more precisely how your algorithm works. Remember, your goal is to communicate. Graders will be instructed to take off points for convoluted and obtuse descriptions. If you cannot solve a problem, give a brief summary of any partial results.

Problem 4-1. Electric Potential of Equally Spaced Charges

Consider a set of n point charges $p_1, p_2, \ldots, p_n$ located at position $1, 2, \ldots, n$ along the x-axis respectively. Each p_i carries an electric charge q_i. This configuration with $n = 7$ is depicted in the following figure:



In this problem we would like to find the electric potential $v(p_i)$ at each point charge p_i induced by all other point charges, given by the formula where c is a constant:

$$v(p_i) = c\left(\sum_{j \neq i}^{n} \frac{q_j}{|i - j|}\right).$$

(a) The induced potential on all the point charges can be computed by multiplying an $n \times n$ matrix $\mathbf{A}$ by the n-dimensional charge vector $\mathbf{q}$. In other words $\mathbf{v} = c\mathbf{A}\mathbf{q}$, where c is the constant in the above formula. Give the matrix $\mathbf{A}$ for $n = 4$. What is $\mathbf{A}$'s form in general?

(b) Give a representation of $\mathbf{A}$ by an $O(n)$-size vector.

(c) Give an algorithm that computes the potential vector $\mathbf{v} = c\mathbf{A}\mathbf{q}$, in $O(n \log n)$ time.

Problem 4-2. Universal Hashing

Recall that a collection $\mathcal{H}$ of hash function from a universe $\mathcal{U}$ to a range R is called **universal** if for all $x \neq y$ in $\mathcal{U}$ we have

$$\Pr_{h \in \mathcal{H}} [\, h(x) = h(y) \,] = \frac{1}{|R|}$$

We want to implement universal hashing from $\mathcal{U} = \{0,1\}^p$ to $R = \{0,1\}^q$ (where $p > q$).

For any $q \times p$ boolean matrix A and any q-bit vector b we define the function $h_{A,b} : \{0,1\}^p \to \{0,1\}^q$ as $h_{A,b}(x) = Ax + b$, where by this we mean the usual matrix-vector multiplication and the usual vector addition, except that all the operations are done modulo 2. For example, if $q = 2, p = 3$ and

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},$$

we have

$$h_{A,b}(x) = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} x_2 + x_3 + 1 & mod\ 2 \\ x_1 + x_3 & mod\ 2 \end{pmatrix}$$

(a) Prove that the collection $\mathcal{H} = \{\, h_{A,b} \,: A \in \{0,1\}^{q \times p}, b \in \{0,1\}^q \,\}$ is universal.

(b) Let $S \subseteq \mathcal{U}$ be the set we would like to hash. Let $n = |S|$ and $m = 2^q$. Prove that if we choose $h_{A,b}$ from $\mathcal{H}$ uniformly at random, the expected number of pairs $(x, y) \in S \times S$ with $x \neq y$ and $h_{A,b}(x) = h_{A,b}(y)$ is $O(\frac{n^2}{m})$.

(c) Let S and T be subsets of $\mathcal{U}$ with size k, where $k = \sqrt{m} = 2^{q/2}$. Give a randomized algorithm for finding $(S \cap T)$ in expected $O(k)$ time and $O(k^2)$ space.

Problem 4-3. Range Query in Binary Search Tree

Given a binary search tree T and a pair of numbers a, b with $a \leq b$, give an algorithm that returns the set of elements in T with key values in $[a, b]$. Your algorithm should run in $O(h + m)$ time, where h is the height of T and m is the number of elements within the range.