ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2019

Problem Set 2

Due date: 23:59, Mar 20, 2019

1. Submit your solutions to Gradescope (www.gradescope.com).
2. In "Account Settings" of Gradescope, set your FULL NAME to your Chinese name and enter your STUDENT ID correctly.
3. If you want to submit a handwritten version, scan it clearly. Camscanner is recommended.
4. When submitting your homework, match each of your solution to the corresponding problem number.

Problem 1: Collecting Toys

There are n types of toys that you wish to collect. Each time you buy a toy, its type is randomly determined from a uniform distribution (i.e., all possible types have equal probabilities). Let $p_{i,j}$ be the probability that just after you have bought your i^{th} toy, you have exactly j toy types in your collection, for $i \geq 1$ and $0 \leq j \leq n$.

(a) Find a recursive equation of $p_{i,j}$ in terms of $p_{i-1,j}$ and $p_{i-1,j-1}$ for $i \geq 2$ and $1 \leq j \leq n$.

(b) Describe how the recursion from (a) can be used to calculate $p_{i,j}$.

Problem 2: Knapsack II

Given n objects and a knapsack, item i weighs $w_i > 0$ kilograms and has value v_i where $n > v_i > 0$. The knapsack has capacity of W kilograms. The numbers n, v_i are integers and w_i, W are real numbers. What is the maximum total value of items that we can fill the knapsack with? Design an efficient algorithm. For comparison, our algorithm runs in $\mathcal{O}(n^3)$.

Problem 3: Counting Friends

There are n students and each student i has 2 scores x_i, y_i. Students i, j are friends if and only if $x_i < x_j$ and $y_i > y_j$. How many pairs of friends are there? Design an efficient algorithm. For comparison, our algorithm runs in $\mathcal{O}(n \log n)$ time.

Problem 4: Equivalent Detection

There are n cards and an "equivalence tester". One can run the tester on any two cards and determine whether they are equivalent or not. The question is: among the collection of n cards, are there more than $n/2$ cards that are all equivalent to one another? Show how to find out the answer with only $\mathcal{O}(n \log n)$ invocations of the tester.

Problem 5: Sequence Merging

You are given three sequences A, B and C. The length of the three sequences is m, n and m+n respectively. In other words, the length of C is the sum of the length of A and B. Design an efficient algorithm to check if A and B can be merged into C such that the order of all the letters in A and B is preserved.

Example 1: A=aabb, B=cba, C=acabbab, then your algorithm should return true.

Example 2: A=aabb, B=cba, C=aaabbbc, then your algorithm should return false.

Problem 6: Polynomial Multiplication

There are two polynomial functions $f = a_0 + a_1x^1 + \cdots + a_{n-1}x^{n-1}$ and $g = b_0 + b_1x^1 + \cdots + b_{n-1}x^{n-1}$. Let h be the product of f and g, i.e., $h = f \cdot g$. When computing h, we need to take the product of terms from f and g respectively. Normally, we define $x^i \cdot x^j = x^{i+j}$. However, in this problem we define $x^i \cdot x^j = x^{i \oplus j}$, where '$\oplus$' is the bitwise XOR operator (`https://en.wikipedia.org/wiki/Bitwise_operation#XOR`).

Suppose $h = c_0x^0 + c_1x^1 + \cdots + c_{n-1}x^{n-1}$. Design an efficient algorithm to calculate the coefficients of h. For comparison, our algorithm runs in $\mathcal{O}(n \log n)$ time.

Hint: Imitate the idea behind the Karatsuba algorithm.