CS240 Algorithm Design and Analysis: Homework #2

Due on October 24, 2017

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 $p_{i,j}$.

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 $\mathcal{O}(n^3)$.

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 friends are there? Design an efficient algorithm. For comparison, our algorithm runs in $\mathcal{O}(n \log n)$ time.

XOR Convolution

Given two arrays $A = a_0, a_1, \cdots, a_{n-1}$ and $B = b_0, b_1, \cdots, b_{n-1}$, return an array $C = c_0, c_1, \cdots, c_{m-1}$ where $c_i = \sum_{j \oplus k = i} a_j b_k$. Design an efficient algorithm. For comparison, our algorithm runs in $\mathcal{O}(n \log n)$ time.

'$\oplus$' is the bitwise XOR operator: `https://en.wikipedia.org/wiki/Bitwise_operation#XOR`

Hint: Define $x^i \cdot x^j = x^{i \oplus j}$, and imitate the Karatsuba algorithm.

DNA Pattern Recognition

There are four possible bases in a DNA sequence: A, G, C, T. Suppose we have two DNA sequences S and P with length n and m where $\sqrt{n} < m < n - \sqrt{n}$. Design an efficient algorithm to find out the minimum number of bases in P that we have to change so that P is a substring of S. For comparison, our algorithm runs in $\mathcal{O}(n \log n)$ time.

For instance, S = “AGCTAGGCTCT”, P = “AAGTCTC”. The answer is 2. We can change P to “TAGGCTC”.

Hint: An application of FFT.

2D Inversions

Given an array of 2D pairs $A = a_0, a_1, \cdots, a_{n-1}$ where $a_i = (x_i, y_i)$, define $a_i > a_j$ as $x_i > x_j$ and $y_i > y_j$.

(a) How many half-inversions are there? a_i and a_j are half-inverted if $i < j$, $x_i > x_j$ and $y_i \geq y' > y_j$ where y' is a fixed constant. Design an efficient algorithm. For comparison, our algorithm runs in $\mathcal{O}(n \log n)$ time.

(b) How many cross-inversions are there? a_i and a_j are cross-inverted if $i < i' \leq j$ and $a_i > a_j$ where i' is a fixed constant. Design an efficient algorithm. For comparison, our algorithm runs in $\mathcal{O}(n \log n)$ time.

(c) How many inversions are there? a_i and a_j are inverted if $i < j$ and $a_i > a_j$. Design an efficient algorithm. For comparison, our algorithm runs in $\mathcal{O}(n \log^2 n)$ time.