CS648A : Randomized Algorithms
Semester II, 2021-22, CSE, IIT Kanpur

Theoretical Assignment 1

Deadline : 11:55 PM, 10th February 2022.

Most Important guidelines

- It is only through the assignments that one learns the most about the algorithms and data structures. You are advised to refrain from searching for a solution on the net or from a notebook or from other fellow students. Remember - **Before cheating the instructor, you are cheating yourself**. The onus of learning from a course lies first on you. So act wisely while working on this assignment.
- Refrain from collaborating with the students of other groups. If any evidence is found that confirms copying, the penalty will be very harsh. Refer to the website at the link: https://cse.iitk.ac.in/pages/AntiCheatingPolicy.html regarding the departmental policy on cheating.

General guidelines

1. This assignment is to be done in groups of 2 students. You have to form groups on your own. You are strongly advised not to work alone.
2. The assignment consists of 2 problems. Each problem carries 50 marks.
3. **Naming the file**:
   The submission file has to be given a name that reflects the information about the type of the assignment, the number of the assignment, and the roll numbers of the 2 students of the group. If you are submitting the solution of Theoretical Assignment x, you should name the file as **Theor_x_Rollnumber1_Rollnumber2.pdf**.
4. **Each student of a group** has to upload the same submission file separately. Be careful during the submission of an assignment. Once submitted, it can not be re-submitted.
5. Deadline is strict. Make sure you upload the assignment well in time to avoid last minute rush.

2-Dimensional Pattern Matching

marks = (10,20,30)

First we shall discuss a different finger printing technique to solve one-dimensional pattern matching problem. The idea is to map any bit string s into a 2×2 matrix $M(s)$, as follows.

- For the empty string ϵ, $M(\epsilon) = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$
- $M(0) = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$
- $M(1) = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$
- For non-empty strings x and y, $M(xy) = M(x) \times M(y)$.

Convince yourself that this fingerprint function has the following properties.

- $M(x)$ is well defined for all $x \in \{0,1\}^*$.
- $M(x) = M(y) \Rightarrow x = y$.

Starting with the two properties of M listed above, solve the following problems.

1. By using the matrix $M(x)$ for a bit-string x, design an algorithm for the pattern matching problem in 1-dimension (text and pattern are bit-strings of length n and m respectively). The algorithm should take $O(n+m)$ time. You may make use of the following assumption.
   Assumption: Any arithmetic operation involving two numbers of arbitrary length can be executed in $O(1)$ time.
2. It can be shown that for a bit string x of length n, entries of $M(x)$ can be quite long; however, they will be bounded by Fibonacci number F_n. Unfortunately, this fact suggests that the assumption mentioned above is not practical. In order to circumvent this problem, we need to work with small numbers - numbers of $O(\log n)$ bits only. Therefore, taking inspiration from one of the lectures, we keep entries of M modulo a prime number p picked randomly uniformly from a suitable range. This is now a practical but randomized Monte Carlo algorithm for 1-dimensional pattern matching. Do proper analysis to find the range from which you need to pick the prime number randomly to achieve error probability $< 1/n^4$.
   (*Most of you might be finding this pattern matching algorithm more complicated than the one we discussed in the class. However, this algorithm has the merit that it can be extended to solve 2-dimensional pattern matching very easily. You will do this extension as the last part of this problem.*)
3. Consider the two-dimensional version of the pattern matching problem. The text is an $n \times n$ bit-matrix X, and the pattern is an $m \times m$ bit-matrix Y. Of course, $m \leq n$. A pattern match occurs if Y appears as a (contiguous) sub-matrix of X. Get inspired from the randomized algorithm for part (b) above to design a randomized Monte Carlo algorithm for this 2-dimensional pattern matching problem. The running time should be $O(n^2)$ and the error probability should be $< 1/n^4$.

Making an intelligent guess

marks=(40,10)

We have a function $F : \{O, ..., n-1\} \rightarrow \{O, ..., m-1\}$. We know that, for $0 \leq x, y \leq n-1$,

$$F((x+y) \bmod n) = (F(x) + F(y)) \bmod m$$

The only way we have for evaluating F is to use a lookup table that stores the values of F. Unfortunately, an Evil Adversary has changed the value of $1/5$ of the table entries when we were not looking. Describe a simple randomized algorithm that given an input z, outputs a value that equals $F(z)$ with probability at least $1/2$. Your algorithm should work for every value of z, regardless of what values the Adversary changed. Your algorithm should use as few lookups and as little computation as possible.

Suppose I allow you to repeat your initial algorithm k times. What should you do in this case, and what is the probability that your *enhanced* algorithm returns the correct answer?