Radomized Algorithms Problemes 2 Fall 2019.

9.- (*) Given a text $T = x_1 x_2, \cdots x_n$ and a pattern $S = s_1 \cdots s_m$, with $m << n$ and both chains over the same alphabet $\Sigma = \{0, 1\}$, we want to determine if S occurs as a contiguous substring of T. For ex., if $T = 10110110010101110$ and $S = 1011$ then $101\boxed{1011}0010\boxed{1011}10$. The standard greedy takes $O(nm)$ steps. There are deterministic algorithms that work in worst-time $O(n + m)$ (Knuth-Morris-Pratt), but they are complicated, difficult to implement and with large implementation constants. The following simple probabilistic algorithm does the job using the fingerprint technique, with a small probability of error. The algorithm computes the fingerprint of S and compares with the fingerprints of successive sliding substring of T, i.e. with $T(j) = x_j \cdots x_{j+m-1}$, for $1 \leq j \leq n - m + 1$.

Matching P, T
Express S as an integer $D(S) = \sum_{i=0}^{m-1} x_{i+1} 2^i$
  so $D(S)$ is a m-bit integer
Choose a prime $p \in [2, \ldots, k]$,
  where $k = cmn \ln(cmn)$, for suitable $c > 1$
Compute $\phi(S) = D(S) \bmod p$
for $j = 1$ to $n - m + 1$ **do**
  Compute $D(T(j)) = \sum_{i=0}^{m-1} x_{j+i} 2^i$
  Compute $\phi(T(j)) = D(T(j)) \bmod p$
  if $\phi(T(j)) = \phi(S)$ **then**
    output match at position j
  endif
endfor

Prove,

(a) This algorithm is one-side, it may output match when there is no match. Prove the $\mathbf{Pr}\,[\textit{output match, when no match}] \leq 1/c$, for suitable $c > 0$.

(b) Prove that the algorithm can be implemented in $O(n + m)$ steps.

10.- How would you modify the previous algorithm so that there is not probability of mismatch?

11.- (MU 2.2) A monkey types on a 26-letter keyboard that has lowercase letters only. Each letter is chosen independently and uniformly at random from the alphabet. If the monkey types 1,000,000 letters, what is the expected number of times the sequence **proof** appears?

12.- An urn contains 11 balls, 3 white, 3 red, and 5 blue balls. Take out 3 balls at random, without replacement. You win 1€, for each red ball you select and lose a 1€, for each white ball you select. Determine the PMF of the amount you win. Give the value of $\mathbf{E}[X]$

13.- (*) (MU 2.9)

(a) Suppose that we roll twice a fair k-sided die with the numbers 1 through k on the die's faces, obtaining values X_1 and X_2. What is $\mathbf{E}[\max(X_1, X_2)]$? What is $\mathbf{E}[\min(X_1, X_2)]$?

(b) Show from your calculation in part (a) that $\mathbf{E}[\max(X_1, X_2)] + \mathbf{E}[\min(X_1, X_2)] = \mathbf{E}[X_1] + \mathbf{E}[X_2]$

(c) Explain why the equation in part (b) must be true by using the linearity of expectations instead of a direct computation.

14.- (*) (MU 2.7) $\mathbb{D}$ Let X and Y be independent geometric random variables, where X has parameter p and Y has parameter q.

(a) What is the probability that $X = Y$?

(b) What is $\mathbf{E}[\max(X, Y)]$?

(c) What is $\mathbf{Pr}[\min(X, Y) = k]$?

(d) What is $\mathbf{E}[X | X \leq Y]$?

15.- (**) (2.18) The following approach is often called **reservoir sampling**. Suppose we have a sequence of items passing by one at a time. We want to maintain a sample of one item with the property that it is uniformly distributed over all the items that we have seen at each step. Moreover, we want to accomplish this without knowing the total number of items in advance or storing all of the items that we see. Consider the following algorithm, which stores just one item in memory at all times. When the first item appears, it is stored in the memory. When the k-th item appears, it replaces the item in memory with probability $1/k$ Explain why this algorithm solves the problem.

The importance of the problem is following : Suppose, you are monitoring a twitter feed and you want to generate a perfectly random sample of k tweets, as that is the maximum you can store in your memory. For instance, this sample can be used for estimating the percentage of tweets on a particular subject.