University of Passau
Faculty of Computer Science and Mathematics

Randomised Algorithms

Winter term 2022/2023, Exercise Sheet No. 8

Hand-out: Mon, 5. Dec.
Hand-in: Sun, 13. Dec.

Dr. Duc-Cuong Dang
Prof. Dr. Dirk Sudholt

Exercise 1 [8 points]

Given three square matrices A, B, C of dimension n of real numbers, we want to detect whether $AB = C$.

(a) If we just use the most basic matrix multiplication and subtraction algorithms, what is the runtime of the exact/deterministic detection?

Consider the following randomized algorithm: it first creates a vector r of length n by independently sampling its entries from $\{-1, 0, 1\}$ uniformly, then computes sequentially $x = Br$, $y = Ax$, $z = Cr$, and finally $t = y - z$. If the entries of vector t are all zeroes, the algorithm outputs an Yes answer (claiming $AB = C$) otherwise the output is No (claiming the opposite).

(b) What is the asymptotic runtime of this randomized algorithm? What kind of mistake, ie. false positive versus false negative or both, can it make? And what type of randomized algorithm is this?

(c) When $AB \neq C$ we want to estimate the probability that the algorithm outputs a wrong answer, ie. claiming that $AB = C$ and we denote this event by $\mathcal{E}$. Define $D = AB - C$, what does event $\mathcal{E}$ mean to the entries of D and $t = Dr$? Use this to show that $\text{Prob}(\mathcal{E}) \leq 1/3$.

Hints: We only need to focus on one non-zero entry of D and the corresponding row in t (why?). The principle of deferred decisions can be used on the entries of r to bound the probability of the event that occurs to the row, eg. think about which entries should be revealed first and which one should be the last.

(d) How can we further reduce this error probability?

Exercise 2 [8 points]

Let h be a hash function that maps objects to s hash values $h_1, \dots h_s$ uniformly at random, that is, given any object x it holds that $\text{Prob}(h(x) = h_i) = 1/s$ for all $i \in \{1, \dots, s\}$. Given a parameter n and a piece of information that s is either n or n^2, we want to find out whether $s = n$ or $s = n^2$. For example, if $n = 2^{16}$, we want to find out whether $s = 2^{16}$ or $s = 2^{32}$.

(a) Use the birthday paradox to construct an algorithm that runs in $o(n)$, ie. in sublinear time, and outputs the correct answers for this task with probability at least $1/2 + \varepsilon$ for some constant $\varepsilon > 0$.

(b) Prove the correctness of the proposed algorithm, that is, the probability of correct answers is at least $1/2 + \varepsilon$ in both cases $s = n$ and $s = n^2$.

Hints: To create k distinct objects, one can use k consecutive integers. Proving the result when $s = n^2$ may require a *lower* bound on the probability of having no collision. Some estimates can be useful for this, such as: $(1 + x)^n \geq 1 + xn$ which holds for all $x \geq -1$ and $n \in \mathbb{N}$, this is called Bernoulli's inequality. You may assume that n is sufficiently large to fulfil inequalities such as $\lceil \sqrt{2n} \rceil + 1 \leq n/2$.

Exercise 3 [4 points]

In this exercise, we look into the argument used in the lecture to show that the modified algorithm Amnesiac gives an upper bound on the expected running time of Gale-Shapley algorithm in finding a stable matching. Such argument is known as *the first order of stochastic dominance* and we will prove it in a more general setting where the random variables can take both positive and negative values.

Let X, Y be random variables that take values in $\mathbb{Z}$ with finite expectations:

(a) Suppose for a moment that X only takes its value in $\mathbb{Z}^+$, explain briefly why $\text{E}(X)$ can be written as the sum $\sum_{i=1}^{\infty} \text{Prob}(X \geq i)$. Express $\text{E}(X)$ in a similar fashion when X only takes value in $\mathbb{Z}^-$.

(b) Use the above to show that if $\text{Prob}(X \geq i) \leq \text{Prob}(Y \geq i)$ for all $i \in \mathbb{Z}$ then $\text{E}(X) \leq \text{E}(Y)$.