University of Zurich
Institute of Computer Science
P.D. Dr. Alexander Souza

Randomized and Online Algorithms

Exercise 1 (John von Neumann's Coin Trick)

Suppose we have a biased coin, which comes up heads with probability p. Use this coin to simulate a fair coin, i.e., one that comes up heads with probability 1/2.
Hint. Look at pairs of flips of the biased coin.

Exercise 2 (Las Vegas Algorithms)

Suppose we are given a Las Vegas algorithm A with success probability $0 < \varepsilon < 1$. Let $T(I)$ denote its running time on input I.
Construct a Las Vegas algorithm A' with success probability 1, i.e., which always returns a solution. Estimate the expected running of A'.
Hint. Recall Geometric distributed random variables.

Exercise 3 (Drunken Sailors)

After a night at the port pub, n sailors return to their ship "slightly" drunk. The ship has n cabins, one for each sailor. Each drunk sailor chooses a free cabin independently and uniformly at random. What is the expected number of sailors that sleep in their own cabin?

Exercise 4 (Neighborings)

We place n people at a round table with $m \geq n$ many chairs, independently and uniformly at random. What is the expected number of people that do not have a direct neighbor?

University of Zurich
Institute of Computer Science
P.D. Dr. Alexander Souza

Spring Term 2018
Sheet 2

Randomized and Online Algorithms

Exercise 1 (Chernoff-Type Bound)

Let $p \in [0, 1/2]$ and let n be a positive integer. For each $i = 1, \ldots, n$ define the random variable X_i, which takes the value $x_i = 2$ with probability p, $x_i = 1$ with probability p, and $x_i = 0$ with probability $1 - 2p$. Let the X_i be independent and define $X = \sum_{i=1}^{n} X_i$.
Apply the method with moment-generating functions and derive a Chernoff-type bound for

$$\Pr [X \geq (1 + \delta) \cdot \mathbb{E} [X]] ,$$

where $\delta \geq 0$ is arbitrary.

Exercise 2 (Load Balancing)

Suppose that there are n jobs which we want to distribute on n computers. We assign the jobs independently and uniformly at random to the computers. Let X_i denote the number of jobs assigned to computer i and let $X = \max\{X_1, \ldots, X_n\}$. Show that

$$\Pr [X > 2 \cdot \log n] \leq \frac{1}{n}$$

for sufficiently large n.
Hint. There is the following version of the Chernoff bound: Let $Y = \sum_{i=1}^{m} Y_i$ for independent indicator variables $Y_i \in \{0, 1\}$ with $\Pr [Y_i = 1] = p_i$. Then $\Pr [Y \geq t] \leq 2^{-t}$ if $t \geq 2 \cdot e \cdot \mathbb{E} [Y]$.

Exercise 3 (Wireless Communication)

In many wireless communication systems, each receiver listens on a specific frequency. A bit b is represented by -1 or $+1$. Noise from nearby other senders can affect the receiver's signal: There are n other senders. Let sender i have strength $0 \leq p_i \leq 1$ and it tries to send bit b_i. The signal obtained by the receiver is

$$S = b + \sum_{i=1}^{n} p_i \cdot B_i.$$

If S is negative, the receiver assumes that b is -1, otherwise $+1$ is assumed for b. We assume that the B_i are independent and uniform random variables, i.e., $\Pr [B_i = +1] = \Pr [B_i = -1] = 1/2$.

(1) Determine the expected value of S and discuss your finding: Is the above protocol plausible?

(2) Give a Chernoff-type bound to estimate the probability that the receiver makes an error in determining b.

University of Zurich
Institute of Computer Science
P.D. Dr. Alexander Souza

Randomized and Online Algorithms

Exercise 1 (Satisfying Assignments)

Let F be a Boolean formula in conjunctive normal form with m clauses for the 3-SATISFIABILITY problem. Prove that there exists an assignment satisfying at least $7/8 \cdot m$ clauses of F.

Exercise 2 (Triangles in Random Graphs)

Consider the random graph model $G_{n,p}$, that is, a graph with n vertices and each of the $\binom{n}{2}$ possible edges exists in the graph with probability p, independently. A triangle is a set of three vertices u, v, w such that the edges edges uv, vw, wu exist in the graph.
Let $p = n^{-c}$ for some constant c. Show the following thresholds for the property "$G_{n,p}$ has a triangle":

(1) $\lim_{n\to\infty} \Pr[G_{n,p} \text{ has a triangle}] = 0$ for $c > 1$, and

(2) $\lim_{n\to\infty} \Pr[G_{n,p} \text{ has a triangle}] = 1$ for $c < 1$.

University of Zurich
Institute of Computer Science
P.D. Dr. Alexander Souza

Randomized and Online Algorithms

Exercise 1 (Randomized Rounding of Integer Linear Programs)

Prove Theorem 5.1 from the lecture: Let $A \in [-\alpha, \alpha]^{n,n}$ be a matrix and let $b \in \mathbb{R}^n$. Let z be any fractional solution for the linear program $Ax = b, x \in [0,1]^n$ (without objective function). Define a vector $X \in \{0,1\}^n$ randomly by

$$X_i = \begin{cases} 1 & \text{with probability } z_i, \\ 0 & \text{otherwise} \end{cases} \quad \text{for all } i = 1, \ldots, n.$$

Then, with high probability, an outcome x of X satisfies

$$Ax - b \in [-c\sqrt{n \log n}, c\sqrt{n \log n}]^n,$$

for a suitable constant $c = c(\alpha) > 0$.

Exercise 2 (Maximum Independent Sets – Revisited)

Recall the Maximum Independent Set problem: We are given a (simple, undirected) graph $G = (V, E)$ on n vertices V and m edges E. The task is to find an independent set S in G having maximum size. An independent set S has the property that, if $u, v \in S$ then $uv \notin E$. The goal of this exercise is to show that the following algorithm is an $O(\sqrt{m})$-approximation:

(1) Formulate Maximum Independent Set as an Integer Linear Program (ILP).

(2) Relax the ILP to an ordinary Linear Program (LP).

(3) Solve the LP optimally and denote the respective solution by x.

(4) If the LP-value is less than $2\sqrt{m}$ return $S = \{v\}$, where v is an arbirary vertex. Otherwise, sample a set S as follows: Any vertex v is included into S with probability $p_v = x_v/\sqrt{m}$, independently.

(5) Correct the set S so that it “becomes” an independent set and return S.

Hint. You may find the arithmetic-geometric-mean inequality useful: $\sqrt{xy} \leq (x + y)/2$ for any non-negative x, y.

University of Zurich
Institute of Computer Science
P.D. Dr. Alexander Souza

Spring Term 2018
Sheet 5

Randomized and Online Algorithms

Exercise 1 (Cow Problem I)

Consider a cow that is standing at an infinitely long fence. The cow knows that the fence has an exit, which is either to its left or right and at least one meter away.
Let OPT denote the direct distance from the cow to the exit and let COW denote the distance of the to-be-designed algorithm the cow will follow to find the exit. The ratio COW/OPT is called cow-petitive factor.
Design a constant cow-petitive deterministic algorithm.
Hint. There exists a deterministic 9-cow-petitive algorithm, in which the cow alternates between left-and-right traversals.

Exercise 2 (Cow Problem II)

If the cow uses randomness in its decision-making, we arrive at the expected cow-petitive factor $\mathbb{E}[\text{COW}]/\text{OPT}$.
Can you design a randomized algorithm with improved expected cow-petitive factor compared to the previous exercise?

Exercise 3 (Online Trading)

Suppose that you have 1 CHF and wish to exchange it to EUR within the next k consecutive days. That is, if you have not exchanged until day k, you will exchange the CHF at any rate. At any day, you can either exchange the entire amount or do nothing at all. Finally, the range of the EUR exchange rate is known to be $[1, M]$, where $M \geq 1$ is also known to you. This means, that the exchange rate r_i of day $i = 1, \ldots, k$, is in the interval $[1, M]$.
Consider the following RESERVATION PRICE strategy: Let $r_i \in [1, M]$ denote rate at day i. If you have not traded yet, if $r_i \geq \sqrt{M}$, trade the entire CHF.
Show that RESERVATION PRICE is a $\sqrt{M}$-competitive strategy.