ETH Zurich
Institute of Theoretical Computer Science
Prof. Dr. Angelika Steger, Prof. Dr. Emo Welzl
Frederik Benzing, Marcelo Matheus Gauy, Pascal Pfister

HS 2017
Graded Homework 1

Randomized Algorithms and Probabilistic Methods

This is the first graded homework exercise set.

Regulations:

- *There will be a total of* ***three*** *special exercise sets during this semester.*
- *You are expected to solve them carefully and then write a nice and complete exposition of your solutions using* ***LaTeX****. Please submit your solutions as a pdf file to ppfister@inf.ethz.ch (filename: [nethz user name]_ghw1.pdf).*
- *You are welcome to discuss the exercises with your colleagues, but we expect each of you to hand in your own,* ***individual writeup****. Write down your name and* ***list all your collaborators*** *in the beginning of your writeup.*
- *Your solutions will be* ***graded****. Each graded homework will account for 10% of your final grade for the course (so 30% of the grade in total).*

Due date: Monday, October 23th, 2017 at 23:59

Solving the following exercises, you may use the following facts without proof:

Fact 1: For all real numbers x it holds that $1 - x \leq e^{-x}$.

Fact 2: For $0 \leq x \leq 1/2$ it holds that $1 - x \geq e^{-2x}$ and $1 - x \geq e^{-x-x^2}$.

Fact 3: For positive numbers $a_1, \ldots, a_m, C, a$ which satisfy $\sum a_i = a$ and $a_i \leq C$ for $i = 1, \ldots, m$, we have $\sum a_i^2 \leq aC$.

Fact 4: Given two non negative integers s, n with $s \leq n$, one can bound the binomial coefficient $\binom{n}{s}$ in various ways. Depending on how large s is compared to n one of the following two bounds (which are far from being tight) is often helpful: We have $\binom{n}{s} \leq 2^n$ and $\binom{n}{s} \leq n^s$. You may also want to use $\binom{n}{n/2} \leq 2^n/\sqrt{n}$ for sufficiently large (even) n.

Exercise 1 *(7 points)*

You are given a bipartite graph $G = (A \dot{\cup} B, E)$ with $|A| = |B| = n$, i.e. each vertex class consists of n vertices. Each vertex $v \in G$ is assigned a fixed collection C_v of colours. An *admissible* colouring c is a proper colouring of the vertices of G (i.e. no two adjacent vertices receive the same colour), such that $c(v) \in C_v$ for all vertices $v \in G$.

Now suppose that for every vertex $v \in G$, we have $|C_v| > \log_2 n + 1$. Show that there exists an admissible colouring.

Exercise 2 *(7 points)*

Let V be a set. Let $\binom{V}{k} := \{S \subseteq V : |S| = k\}$ be the set containing all subsets of V of size k. A k-uniform hypergraph H is a pair (V, E), where V denotes the set of vertices, and $E \subseteq \binom{V}{k}$ denotes the set of (k-uniform hyper)edges (each edge corresponds to a subset of vertices of V with exactly k elements). A random k-uniform hypergraph $H = H^k(n, p)$, is a random hypergraph on n vertices where each (k-uniform hyper)edge is added independently with probability p to the hypergraph H.

Let $H \sim H^3_{n,p}$ be a random 3-uniform hypergraph on n vertices. Show that $p = 2\ln(n)/n^2$ is a sharp threshold for the property 'H has no isolated vertices'.

Exercise 3 *(5+8 points)*

Let G be graph on n vertices, and let c be an edge-colouring of G which is *proper*, i.e. no two edges which share a vertex have the same colour.

A subgraph H of G is called rainbow, if its edges have pairwise distinct colours. For a subset of vertices $A \subset V(G)$, we denote by $G[A]$ the subgraph of G which is induced by A. That is, $G[A]$ has vertex set A and contains precisely the edges from G which have both endpoints in A.

1. Show that, for n sufficiently large, we can find a set $A \subset V(G)$ of at least $n^{1/4}$ vertices, so that $G[A]$ is rainbow.
2. We say that the colouring c is $\alpha(n)$-bounded if no colour of c appears on more than $\alpha(n)$ edges. Assume that c is not only proper but also $n/(\log n)^3$-bounded. For two subsets of vertices $A, F \subset V(G)$, denote by $H(A, F)$ the subgraph of G that consists of all edges that are incident to A but not to F[1] (and has vertex set $V(G)$). Show that, for some sufficiently large absolute constant C and sufficiently large n, there are *disjoint* sets A and F, so that $|A| \geq \log n$, $|F| \leq Cn/\log n$ and so that $H(A, F)$ is rainbow.

[1] An edge is incident to a subset $S \subset V(G)$ if at least one of its endpoints lies in S.

Exercise 4

(2+5+2+5 points)

Suppose you have a friend, Paul, who likes boardgames. He recently discovered a classic from the 70s called Mastermind and invites you and Alice, another friend of yours - she likes riddles - to play a round or two. You accept the invitation, not because you like boardgames (you don't) but because you like money and it is an old habit of Paul's to give 10 dollars to the best player of the evening.

Since Alice and you are clever, the game gets boring rather quickly. Fortunately, Paul comes up with the following interesting adaption of the game: Paul secretly chooses a codeword $a = (a_1, \dots, a_n)$, where $a_1, \dots, a_n \in [n]$. Your task is to find out the codeword. In order to do so, you make guesses of the form $g = (g_1, \dots, g_n)$, with $g_1, \dots, g_n \in [n]$. Paul then tells you the number of positions $i \in [n]$ for which $a_i = g_i$. He calls this number $t(a, g)$. Your aim is to (uniquely) identify Paul's codeword with as few guesses as possible. We will henceforth refer to the potential entries $1, \dots, n$ at each position i as colours.

Alice is taking the RandAlg course and decides to make random guesses. She does this for some time and suddenly makes a guess g for which $t(a, g) = 0$, we call such a guess a *zero-guess*. She realizes that this guess lets her 'eliminate one colour from each position'. More precisely, for each position i she knows that $a_i \neq g_i$ and she can eliminate g_i from the list of possible colours for position i. So she comes up with the following strategy:

She keeps a list of possible colours L_i for each position i. She starts with $L_i = [n]$ for all $i = 1, \dots, n$. Given these lists she makes a random guess $g = (g_1, \dots, g_n)$ as follows: She chooses each g_i uniformely and independently at random from L_i. She repeats this until she makes a zero-guess $(g_1, \dots, g_n)$, given this guess she deletes g_i from L_i for all positions i. After $n - 1$ zero-guesses, there is only one possible colour remaining at each position and Alice knows Paul's codeword.

1. What is the expected number of guesses Alice needs to make until she knows the codeword (if she follows the strategy given above)? You don't need to give a closed-form expression. Show that this expectation E is exponential in n, i.e. $E = \Omega(c^n)$ for some $c > 1$.

Alice's strategy is taking forever (it is almost midnight, and she hasn't finished yet), and you figure out that you should be able to do better. You also don't want anything as complicated as Alice's list-keeping (remember, you don't even like boardgames, so why bother with lists?). This inspires you to solve the following two exercises.

2. Show that $Cn \ln n$ guesses taken independently of each other and uniformly at random suffice whp to determine Paul's codeword. Here C is a sufficiently large constant independent of n.
3. Use 2. to describe a strategy that determines Paul's codeword in expected time $O(n \ln n)$. [Note: You may attempt this question without having solved 2.]

Even though you are really getting tired, Paul poses one more challenge. Suppose that at each of the n positions you have only 2 possible colours, i.e. Paul now chooses a codeword $a = (a_1, \dots, a_n)$ where $a_i \in \{1, 2\}$ for each $i = 1, \dots, n$. As before, you can take guesses $g = (g_1, \dots, g_n)$, where now $g_i \in \{1, 2\}$ for all $i = 1, \dots, n$ and Paul reports $t(a, g)$ the number of positions in which you guessed correctly.

4. Show that you can determine Paul's codeword with at most $Cn/\log n$ guesses whp, where again C is a sufficiently large constant independent of n.