Exam problems for the course 'Randomized Algorithms' (DM839)

Department of Mathematics and Computer Science
University of Southern Denmark

April 9, 2014

The problems are handed out Thursday April 7, 2014. The solutions must be returned by Wednesday May 14, 2014 at 10.00. You should place 2 copies of your report including programs and relevant output from these in Jørgen Bang-Jensen's mailbox in the secretary's office. You are allowed to work in groups (of at most 3 students) and make a single hand-in per group. remember to write the names of all members of the group. The report should be made using LaTeX or similar. Handwriting is not acceptable.

It is important that you explain how you obtain your answers and argue why they are correct.

You may choose to solve and hand in any subset of the exercises whose points sum to 60, with the restrictions that if you hand in Problem 1B (4B) then you should also hand in Problem 1A (4A) (both pairs summing to 20 points in total). If you hand in answers to problems whose points sum to more than 60 points, I will only correct a random subset summing to 60 points. The final grade for the course will be based on an overall impression of your performance on this set of problems and the oral exam on June 25.

The evaluation of your report will put weight on simple short precise answers/documentation. For the programming subproblems, documentation includes

- a precise description of the purpose of the experiment
- a well motivated design of the experiment
- a clear description of the outcome of the experiment, and
- a careful analysis and discussion of the outcome compared to theoretical predictions.

For the experiments you will need a source of random / pseudo-random numbers. In your report, you must explain carefully what source you have used.

PROBLEM 1A (10 point)

This problem considers robustness of a score (grade) given to students based on a multiple choice test. The problem set-up is based on the paper Frandsen and Schwartzbach, *A singular choice for multiple choice,*
`[http://dl.acm.org/citation.cfm?id=1189164]`.

Assume a multiple choice test consists of n questions, each having 4 choices. For each question precisely one choice is correct. Students are allowed to make 0 or 1 "check" (cross) for each question. The score for a question is 1 if the student has checked the correct choice, $-\frac{1}{3}$ if the student has checked a wrong choice and 0 if no choices are checked. The score for the test is computed as the sum of the scores for all questions. The maximal score is therefore n. We assume that the test is used only to decide pass/fail, and the threshold for passing is a 50% score, i.e. a score $\geq \frac{n}{2}$.

Define a **challenged** student to be a student that knows the answers to at most 40% of the questions.

Let us assume that a challenged student leaves no questions unanswered. Then clearly he has nothing to lose by guessing the answers to the questions he does not know. So assume that he accordingly puts down checks at uniformly random choices (one per question).

Define a multiple choice test to be **good**, if the probability that a challenged student passes is at most 5%.

A teacher has to make a test, and naturally he wants it to be good. He suspects that if he has enough questions in the test then it will be good. This is indeed correct.

In this problem you are required to find a (small) size n of the test that ensures it is good. You are required to do this both theoretically using Chernoff bounds (see Mitzenmacher and Upfal, Chapter 4) and experimentally with high confidence.

Assume the challenged student guesses $m = \frac{3}{5}n$ questions and define

$$X_i = \begin{cases} 1, & \text{if the } i\text{th guess is correct} \\ 0, & \text{otherwise} \end{cases}$$

Define $X = \sum_{i=1}^{m} X_i$

1.1 Subproblem Determine $E[X]$.

1.2 Subproblem Show that the challenged student only passes if $X \geq \frac{3}{2}E[X]$.

1.3 Subproblem Using the Chernoff bound technique, determine a size n of the test for which the challenged student only passes with probability at most .05.

The teacher suspects that a much smaller n than the one resulting from the Chernoff bound really suffices to make the test good. He decides to run an experiment, where he simulates challenged students making guesses. For a candidate value n, he simulates R challenged students. Define the following notation

$$Y_{n,i} = \begin{cases} 1, & \text{if the } i\text{th simulated challenged student passes the test} \\ & \text{with } n \text{ questions} \\ 0, & \text{otherwise} \end{cases}$$

Let $p_n = \Pr(Y_{n,i} = 1)$, i.e. the test is good precisely when $p_n \leq .05$. Let $Y_n = \sum_{i=1}^{R} Y_{n,i}$.

1.4 Subproblem Determine $E[Y_n]$.

The teacher decides to accept a test (that is, the proposed size of n) as good, if the outcome Y_n satisfies that $Y_n \leq 0.04R$.

1.5 Subproblem Determine a value of R that is sufficiently large to ensure that a non-good test is accepted as good with probability at most 0.05. You should use the Chernoff bound technique. Hint: Recall that we often used the stronger version of Chernoff bounds which is given in exercise 4.7 in MU. You may use this version without proving its validity.

PROBLEM 1B (10 point)

This is the programming part of the first problem.

1.6 Subproblem You should implement a method that given n, R uses a random/pseudorandom source to determine an experimental value for Y_n.

1.7 Subproblem Using that method you should construct and implement an algorithm that determines a (small) n that ensures a good multiple choice test.

1.8 Subproblem Analyze what confidence you can have in the outcome from running your algorithm above.

1.9 Subproblem Run the algorithm and compare the outcome of the experiment with the earlier theoretically determined minimal size of a good test.

PROBLEM 2 (10 point)

This problem concerns the method of deferred decisions. The goal is to show that the random graph $G_{2\ell,p}$ with $p = \frac{1}{2}$ has a perfect matching with probability at least $\frac{1}{3}$.

Let the vertices of $G_{2\ell,\frac{1}{2}} = (V, E)$ be labelled $v_1, \ldots, v_{2\ell}$ and consider the following approach, where we try to find a perfect matching by repeatedly matching the lowest indexed unmatched vertex to some other (unmatched) vertex using the following algorithm:

1. Set $M = \emptyset$ and $S = V$
2. For $i := 1$ to ℓ do

   (2.1) Let $j(i)$ be the smallest index of a vertex in S. Find the set $N_i = \{y | y \in S \text{ and } v_{j(i)}y \in E\}$ (the neighbours of $v_{l(i)}$ which are still unmatched).

   (2.2) If $N_i \neq \emptyset$ then take a vertex $y_i \in N_i$ and let $M := M + v_{j(i)}y_i$; else let $y_i \in S$ be arbitrary.

   (2.3) Set $S = S - v_{j(i)} - y_i$.

Let A_i be the event that $N_i = \emptyset$ (that is we fail to match $v_{j(i)}$ in iteration i).

Question a:

Show that $Pr(A_i) = 2^{-2(\ell-i)-1}$.

Question b:

Explain how you used the principle of deferred decisions to determine the probability of A_i. Hint: Note that for each pair x, z of vertices, we examine whether there is an edge between these at most once during a run of the algorithm. Does the probability of A_i depend on what happened in earlier steps (lower number than i)?.

We want to estimate the probability of the event $A = \bigcap_{i=1}^{\ell} \bar{A}_i$, that is the probability that none of the events A_i occur and hence we find the desired perfect matching.

Question c:

Show that $Pr(A) \geq \frac{1}{3}$

PROBLEM 3 (10 point)

This problem is about the color-coding technique of Alon et al. which we have seen in the course.

Question a:

Give a short explanation in your own words of how the color-coding technique for checking the existence of a directed path of length k in a digraph works. This includes the main steps, expected complexity and how to derandomize the method, and which complexity the resulting algorithm will have. You do not have to prove the results, just describe them in enough detail.

Question b:

Explain how one can use the color-coding technique to check whether a digraph contains two directed cycles C_1, C_2 both of length k such that $|V(C_1) \cap V(C_2)| = 1$, that is, the cycles share exactly one vertex. You should give the complexity of your algorithm (expected running time). You should also explain how to derandomize your algorithm.

Problem 4A (10 point)

Let $G = (V, E)$ be an undirected graph with no parallel edges. An **orientation** of G is a digraph $D = (V, A)$ which we can obtain from G by replacing each edge $uv \in E$ by one of the arcs $u \to v$, $v \to u$.

Let $D = (V, A)$ be on orientation of an undirected graph $G = (V, E)$. A ordered triple $x, y, z \in V$ of distinct vertices of D is **in-bad** if A contains the arcs $x \to y$, $z \to y$ but there is no arc between x and z. Similarly, an ordered triple $x, y, z \in V$ of distinct vertices of D is **out-bad** if A contains the arcs $y \to x$, $y \to z$ but there is no arc between x and z.

For a given undirected graph $G = (V, E)$ we say that it has an **in-good** respectively an **out-good** orientation if there exists an orientation $D = (V, A)$ of G which no triple of vertices is in-bad, respectively is out-bad. Finally, we say that G has a **good** orientation if there is an orientation $D = (V, A)$ of G which is both in-good and out-good.

For a given input graph $G = (V, E)$ we denote by $\mathcal{E}(G)$ be the set of all triples (x, y, z) of G with the property that $xy, yz \in E$ and $xz \notin E$. For example, if G is just a 4-cycle $v_1v_2v_3v_4v_1$, then $\mathcal{E}(G)$ contains the 4 triples $(v_1, v_2, v_3), (v_2, v_3, v_4), (v_3, v_4, v_1), (v_4, v_1, v_3)$. Note also that this graph does have an in-good orientation, namely $v_1 \to v_2 \to v_3 \to v_4 \to v_1$ and this orientation is also out-good.

Consider the following randomized algorithm $\mathcal{A}$ for constructing an in-good orientation of a given graph $G = (V, E)$ with m edges (the value of α is to be determined later):

1. Randomly orient every edge $uv \in E$.
2. Repeat up to $2\alpha m^2$ times, terminating if there is no in-bad triple left:
   2.1 Chose an arbitrary in-bad triple $(x, y, z) \in \mathcal{E}(G)$
   2.2 pick one of the two edges of the triple and reorient it (e.g. change $x \to y$ to $y \to x$).
3. If an in-good orientation has been found, return it.
4. Otherwise, return that G has no in-good orientation.

Question a:

Suppose the input graph G with m edges does have an in-good orientation. Prove that the expected number of reorientation steps until the algorithm finds an in-good orientation is at most $O(m^2)$. Hint: consider some in-good orientation $D^* = (V, A^*)$ and measure how far the current orientation $D' = (V, A')$ is from agreeing with D^*.

Question b:

Explain why the algorithm $\mathcal{A}$ always return the correct answer if G does not have an in-good orientation.

Question c:

Which value of α should we choose if we want the error probability of $\mathcal{A}$ to be at most 2^{-10}? You must prove your claim.

PROBLEM 4B (10 point)

This continues Problem 4A.

Question d:

Explain how to modify the algorithm $\mathcal{A}$ to an algorithm $\mathcal{B}$ which with high probability will return a good orientation of the given input graph G provided it has such an orientation. What does the modification imply for your analysis of the expected running time? Does it get tighther or looser?

Question e:

Prove that every graph $G = (V, E)$ has an orientation $D = (V, A)$ for which at least half of the triples in $\mathcal{E}(G)$ are in-good and describe a simple algorithm for finding such an orientation. Hint: consider an appropriate acyclic orientation.

PROBLEM 5 (20 point)

Implement the processes 1. and 2. of the exploratory assignment on pages 124-125 in the coursebook and answer the corresponding questions 1. and 2. with the following changes:

- You do not have to prove the claimed probability in 2., but you should do so in 1. Hint for 1.: look at small values of n and try to prove a lower bound for the number of different nodes that must be sent before the tree can become fully marked. Use this bound and an analogy to another problem that we have studied to prove that the expected number of nodes sent will be $\Omega(N \log N)$.
- Instead, for 2. you should give a theoretical upper bound on the number of unmarked vertices just before the last node that was sent made the whole tree marked. Compare this bound to the experimental values you find (collect this information).

For 1. and 2. you should report the number of nodes send as well as other useful information as requested in the text on page 125.

PROBLEM 6 (10 point)

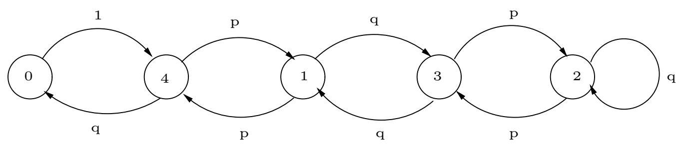

Figure 1: A Markov chain. The states are drawn in the order shown to make the drawing look nicer. The numbers $1, p, q$ are transition probabilities so $p + q = 1$.

I have 4 umbrellas, some at home, some in the office. I keep moving between home and office. I take an umbrella with me only if it rains. If it does not rain I leave the umbrella behind (at home or in the office). It may happen that all umbrellas are in one place, I am at the other, it starts raining and I must leave, so I get wet. Assume now that I have been using this strategy (take an umbrella if it rains, otherwise not) for a long time. I want answers to the following questions.

1. If the probability of rain is p, show that the probability that I get wet is given by $\frac{p(1-p)}{5-p}$.
2. Suppose the probability of rain is $p = 0.6$. What is the probability that I will get wet if I have 4 umbrellas?
3. With the probability of rain being $p = 0.6$ how many umbrellas should I have so that, if I follow the strategy above, the probability I get wet is less than 0.01?

To answer my questions/claims, consider a Markov chain taking values in the set $S = \{i : i = 0, 1, 2, 3, 4\}$, where $i \in S$ represents the number of umbrellas in the place where I am currently at (home or office).

Question a:

Explain why the digraph in Figure 1 represents a Markov chain model for the problem.

Question b:

Find the stationary distribution $(\pi_0, \pi_1, \pi_2, \pi_3, \pi_4)$ and use this to answer my first and second question.

Question c:

Generalize the Markov chain above to states $0, 1, 2, \ldots, N$, write up the stationary distribution and show that the probability of getting wet when I have N umbrellas is given by $\frac{p(1-p)}{N+1-p}$. Use this to answer my third question.

Question d:

Explain what happens if we take $p = 1$ and why that is not an error of the model.