Institut for Matematik og Datalogi
Syddansk Universitet, Odense

10 November 2023
JBJ

DM551-MM851 – 2. Exam assignment

Hand in by Monday December 4 at 09:00.

Rules

This is the second of two sets of problems which together with the oral exam in January constitute the exam in DM551/MM851. **This second set of problems must be solved individually.** Any collaboration with other students will be considered as exam fraud. Thus you are not allowed to show your solutions to fellow students. On the other hand, you can learn a lot from discussing the problems with each other so you may do this to some extend, such as which methods can be used or similar problems from the book or exercise classes.

Remember that this counts as part of your exam, so do a good job and try to answer all questions carefully. It is important that you **argue so that the reader can follow your calculations and explanations**.

How to hand in your report

Your report, which should be written in english, must be handed in on itslearning by Monday December 4, 2023 at 9:00.

On the first page you must write your **name** and the first 6 digits of your **CPR-number**. Do not write the last 4 digits!

Problems

Solve the following problems and **Remember to justify all answers:**

Problem 1 (15 points)

This problem is about random permutations of a set with n distinct elements. For simplicity we assume that $n = 2^k$ for some $k > 0$. Suppose we have an algorithm $\mathcal{A}$ which given a set X of $2r$ elements generates a random subset X_1 of size r. We want to use this to make an algorithm RANDPERM which takes a set S of 2^p vertices and returns a random permutation of S. RANDPERM will do this as follows:

- If $p > 0$ then first use $\mathcal{A}$ to find a random subset X_1 of S of size 2^{p-1}. Then call RANDPERM recursively on X_1 and $X_2 = S \setminus X_1$ respectively to get permutations π_1 and π_2 of X_1 and X_2 respectively and return the permutation $\pi = \pi_1\pi_2$.
- If $p = 0$, that is $S = \{x\}$ for some x, just return x (the permutation with just that element).

For example, if $n = 4$ and $X = \{a, b, c, d\}$ the algorithm $\mathcal{A}$ may select $\{a, c\}$ as X_1 (so $X_2 = \{b, d\}$) and the two recursive calls to RANDPERM may result in the permutations $\pi_1 = ca$ and $\pi_2 = bd$, respectively so the algorithm would return the permutation $\pi = \pi_1\pi_2 = cabd$.

Question a:

Prove by induction on p that if $\mathcal{A}$ works correctly, then RANDPERM returns a random permutation of any given set S of $n = 2^p$ elements.

Question b:

Describe an implementation of the algorithm $\mathcal{A}$ and prove that this will return a random subset of size $\frac{S}{2}$ of S when the input is a set S with an even number of elements.

Problem 2 (10%)

Solve the recurrence equation $a_n = a_{n-1} + 2a_{n-2}$ with initial conditions $a_0 = 4$ and $a_1 = 6$.

Problem 3 (15 %)

Consider the following variant of the coupon collector problem where the number of distinct coupons n is an even number: Each time you collect two coupons. If they are both new (not among those you already have) you can keep them. If at most of them is new, you cannot keep any of them and hence make no progress towards the goal which is still to collect all n different coupons.

Question a:

Suppose you have i of the coupons already for some even number $0 \leq i < n$. What is the probability of having $i + 2$ coupons after the next step?

Question b:

Prove that when you have exactly $i < n$ different coupons, for some even integer i, the expected number of times you have to collect two coupons before you will have $i + 2$ coupons is $\frac{\binom{n}{2}}{\binom{n-i}{2}}$.

Question c:

Show that the expected number of coupons you need to collect before you have all coupons is $O(n^2)$. You may use that $\frac{1}{q(q-1)} = \frac{1}{q-1} - \frac{1}{q}$.

Problem 4 (15 points)

In Kleinberg and Tardós section 13.4 you saw a randomized approximation algorithm $\mathcal{B}$ for MAX-3-SAT whose expected number of satisfied clauses is within a factor $\frac{7}{8}$ of optimal. In fact we showed that if there are m clauses in the input, then the expected number of clauses that will be satisfied by $\mathcal{B}$ is $\frac{7 \cdot m}{8}$ and from this we proved (using the probabilistic method) that every instance of 3-SAT on m clauses has a truth assignment which satisfies at least $\frac{7 \cdot m}{8}$ clauses. We also saw that the expected number of times we need to guess a truth assignment (run $\mathcal{B}$) before obtaining a truth assignment which satisfies at least $\frac{7 \cdot m}{8}$ clauses is less than $8m$.

Below we want to use the same idea to find a randomized approximation algorithm for MAX-k-SAT, where $k \geq 3$.

Question a:

What is the expected number R of clauses we will satisfy if there are m clauses? You must show how to obtain your answer.

Question b:

Prove that there exists a truth assignment that makes at least R of the clauses true.

Question C;

Describe a Las Vegas algorithm $\mathcal{A}$ which given a k-SAT formula $\mathcal{F}$ will find a truth assignment that makes at least R of the clauses in $\mathcal{F}$ true.

Question D:

Give an upper bound on the expected running time of $\mathcal{A}$. Hint: follow the idea in Kleinberg and Tardós section 13.4.

Problem 5 (20 %)

Consider an experiment where we randomly distribute n balls into m distinct boxes labelled $B_1, B_2, \ldots, B_n$. Assume below that $n = m^2$.

Question a:

Prove that the expected number of balls in a fixed box B_i is m.

Question b:

Use Chebyshev's inequality to bound the probability that the number of balls in box B_i is more than 50% away from its expected value.

Question c:

Use the Chernoff bounds (13.42) and (13.43) from Kleinberg-Tardós to bound the probability that the number of balls in box B_i is more than 50% away from its expected value.

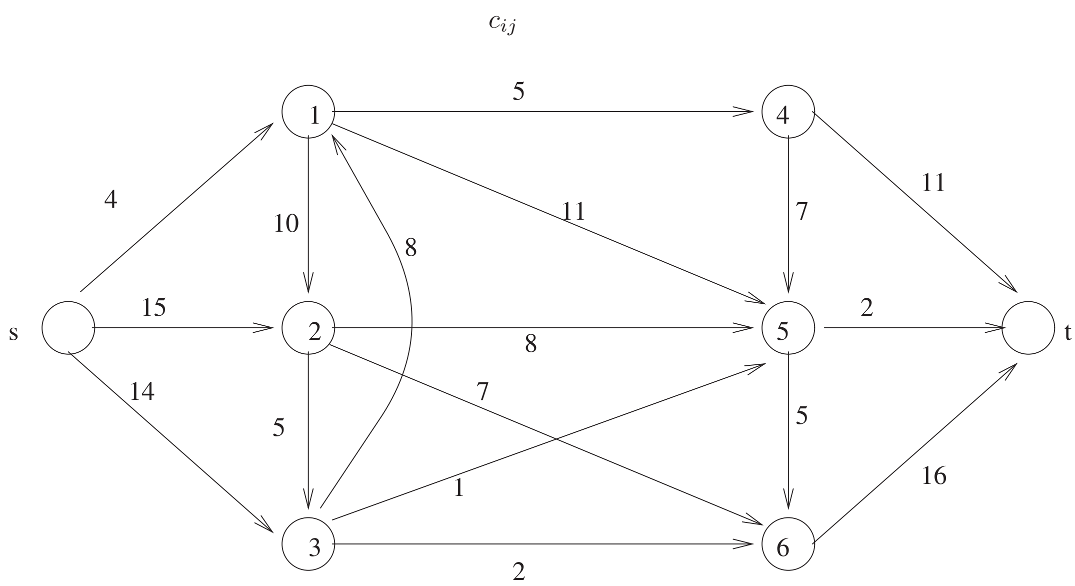
Question d:

Compare the two bounds above and explain the difference. Which bound is best as m gets large?

Question e:

Use your bound in Question c and the union bound to show that when $m = 100$ the probability that there is any box which has less than 50 or more than 150 balls is less than 1%.

Problem 6 (25 point)



Figur 1: The network $N = (V, A, c, s, t)$. The value of the capacity function c is show along each arc.

Question a:

Give a short description of the Edmonds-Karp algorithm for finding a maximum flow and illustrate the algorithm on the network in Figure 1. Remember to say how much you augment by along each path. **In order to make correction easier you should do this exactly as follows**: each new augmenting path should not only be a shortest path in the current residual network, it should also have the smallest name lexicographically. That is, the path $s14t$ is the first augmenting path.

You should draw the residual network each time when there are no more augmenting paths of the current shortest path length. Thus after listing a set of augmenting paths of length 3 (according to the rule above) so that no more augmenting paths of length 3 can be found (in the current residual network), you give the current residual network and then go on to the next set of paths (which will have length 4, as you will see).

Question b:

Give the values on every arc of the resulting maximum flow f^*, give its value and also a minimum cut whose capacity shows that f^* is a maximum flow.

Question c:

Suppose now that we increase the capacity of the arc from vertex 5 to t to 12. Use your final residual network above to say what the value of a new maximum flow will be and give a new cut that shows that this new value is indeed the maximum.