Algorithms Exam [1]

Oct. 20 2011
kl 14 - 18
“väg och vatten” salar’

Ansvarig:
Devdatt Dubhashi Tel. 073 932 2226 Rum 6479 EDIT

Points :	60	
Grades:	Chalmers	5:48, 4:36, 3:28
	GU	VG:48, G:28
	PhD students	G:36
Helping material :	course textbook, notes	

- Recommended: First look through all questions and make sure that you understand them properly. In case of doubt, do not hesitate to ask.
- **Answer all questions in the given space on the question paper (the stapled sheets of paper you are looking at). The question paper will be collected from you after the exam. Only the solutions written in the space provided on the question paper will count for your points.**
- Use extra sheets only for your own rough work and then write the final answers on the question paper.
- Try to give the most efficient solution you can for each problem - your solution will be graded on the basis of its corectness *and* efficiency – a faster algorithm will get more credit than a slower one. In particular a brute force solution will not get any credit.
- Answer concisely and to the point. (English if you can and Swedish if you must!)
- Code strictly forbidden! Motivated pseudocode or plain but clear English/Swedish description is fine.

Lycka till!

[1] 2011 LP 1, DIT600 (GU) / TIN092 (CTH).

Problem 1 Nuts and Bolts [10] You open a IKEA furniture box and find a plastic bag containg a set of n nuts of different sizes and the matching bolts. However they are all mixed up in the bag. You need to find the smallest nut and its corresponding bolt. However the only operation you are allowed is to test a nut against a bolt: either they match or you discover one is too big and the other too small. Show how to find the smallest nut and corresponding bolt with at most $2n - 2$ testing operations. Give a brief justification for both correctness and efficiency.

Problem 2 Changing cost in MSTs [10] Consider a network of computers represented by a graph $G = (V, E)$: the vertices are computers and an edge represents a communication link between the two endpoints. The links are owned by different service providers, and each edge e has a number c_e associated with it which is the cost they charge for using it. To ensure communication in the network, you have constructed a minimum cost spanning tree T. Now one of the service providers want to increase the cost of one specific edge e^* in the tree T, but they don't want to risk losing a customer. So they would like to figure out the maximum amount by which they can increase the cost of e^* that will still keep it as part of the MST T. Design and analyse an efficient algorithm to find the maximum increase that is possible for this.

(a) [2 pts] What are the other edges that could replace e^* if it becomes too expensive? (Describe them in terms of a graph theoretic concept we studied in the course.)

(b) [3 pts] Give an algorithm to solve the problem that runs in time $O(|E|)$. You can give a high level description of the algorithm here and a more detailed one in part (c) below.

(c) [3pts] Argue that your algorithm is correct. (State properties of MSTs you need)

(d) [4 pts] Justify the running time with the use of appropriate data structures and algorithms. (You may assume that both the graph G and the tree T are represented by a list for each vertex of its neighbours in G and T respectively.

Problem 3 Crowdsourcing [10] With the advent of Web 2.0, a popular technique to co-ordinate the use of human intelligence to perform tasks is *crowdsourcing*. For example, in *Amazon's Mechanical Turk*, "Requesters" are able to post tasks known as HITs (Human Intelligence Tasks), such as choosing the best among several photographs of a store-front, writing product descriptions, or identifying performers on music CDs. "Workers" can then browse among existing tasks and complete them for a monetary payment set by the "Requester".

In this problem we consider a simple model of such a system. You are a "Requester", you have a yes/no question to which you need an answer, and there are n "Workers" you can ask. Each "Worker" you ask answers and then you aggregate the answer (for example by majority vote). "Worker" i charges a fee of SEK c_i and has a reliability of r_i (estimated from past experience). You have a total budget of SEK B which you can use to pay the "Workers" you ask to collect an answer for the question.

Your task is to select a subset S of the "Workers" to ask so that their total cost is within your budget: $\sum_{i \in S} c_i \leq B$ and the total reliability $\sum_{i \in S} r_i$ is as large as possible.

(a) [2 pts] Consider the greedy rule: pick the cheapest "Workers" first. Give a simple counter-example to show that this rule doesn't work.

(b) [2 pts] Consider the greedy rule: pick the most reliable "Workers" first. Give a simple counter-example to show that this rule doesn't work.

(c) [2 pts] Consider the greedy rule: pick the "workers" in decreasing order of $\frac{r_i}{c_i}$. Give a simple counter-example to show that this rule doesn't work.

(d) [4 pts] Suppose we can pay a worker fractionally i.e. we pay only for a fraction $0 \leq x_i \leq 1$ of the time of "Worker" i: so the total cost paid is $\sum_{1 \leq i \leq n} c_i x_i \leq B$ and the total reliability is $\sum_{1 \leq i \leq n} r_i x_i$. Argue that the greedy rule in part (c) produces an optimal solution in this case.

Problem 4 Crowdsourcing, take two [10] Return to the crowdsourcing problem, given in Problem 3 (without the modification in Problem 3 (d)). Let $OPT(i, K)$ be the maximum reliability you can get from "Workers" $1 \cdots i$ and total budget K.

(a) [1 pt] In this notation, what is the final solution we want?

(b) [1 pt] What is the value of $OPT(0, K)$ and $OPT(i, 0)$? (Assume all costs are strictly positive - nobody works for free!).

(c) [3 pts] Write a recurrence for $OPT(i, K)$. (HINT: consider whether or not to ask "Worker" i.)

(d) [2 pts] Using (b) and (c), implement the recurrence efficiently in pseudocode.

(e) [1 pt] What is the time and space complexity of your algorithm? Is it a polynomial time algorithm?

(f) [2 pts] How do you actually find the set of "Workers" to answer the question?

Problem 5 Second closest point [10] In class and in the book we discussed a $O(n \log n)$ algorithm to find the distance between the closest pair among n points in the plane. In this problem you are asked to develop a $O(n \log n)$ time algorithm for the distance between the *second*-closest pair of points. (Assume that the distance between each pair of points is unique, so there are $\binom{n}{2}$ different distances and the closest and second closest pairs are therefore unique.)

(a) [2 pts] Describe the "Divide" step, and show what the recursive calls are.

(b) [3 pts] Describe the "Conquer" or "Combine" step: how to combine the solutions to the subproblems described in (a) to obtain a solution to the full problem. Describe carefully the specific things your recursive calls should return and how you will use them for the combine step.

(c) [4 pts] Give a brief justification for the correctness of the algorithm.

(d) [1 pts] Write a recurrence for the running time for the algorithm and state the solution to it.

Problem 6 Planning Chalmers Masters programmes [10] You are in charge of planning allocation of students to Chalmers masters programmes. There are a total of m students applying to the n different Chalmers masters programmes. Each student has a list of the masters programems they would be willing to join. To complete a degree they need to do a masters project at one of p different local companies. Each company has a list of masters programmes they are willing to take students from e.g. Volvo can take from mechanical engineering, computer science or industrial engineering whereas AstraZeneca can take from chemical and biological engineering and computer science. Masters programme i has a maximum limit of a_i students it can admit and company j has a maximum limit of b_i interns they can take.

(a) [4 pts] Construct a network so that by computing a maximum flow on it, you can find the maximum number of students who can be admitted to Chalmers and how they can be allocated to the different masters programmes and companies.

(b) [3 pts] Give a correctness argument for the approach described in (a).

(c) [3 pts] What is the running time of your algorithm?

Problem 7 Please Take Me Out Of Here [10] We revisit the underlying problem of the course labs (a variant of the "Find Sophie" Facebook puzzle).

Given a graph on n vertices with distances $d(u, v) \geq 0$ between any two vertices, a special vertex $s \in V$ and a probability distribution $p_v, v \in V$ on vertices ($\Sigma p_v = 1$), the problem was to find a path $P = v_1, v_2, \cdots, v_n$ in G, with $v_1 = s$, such that the expected search time $E[P] := \sum_i p_{v_i} \sum_{j<i} d(v_j, v_{j+1})$ is minimized.

Consider the decision version of the problem: given the data above and a (rational) value a, is there a path P, starting in s, with $E[P] \leq a$?

(a) [4 pts] Show that the decision problem is in $\mathcal{NP}$.

(b) [4 pts] Show that the problem is $\mathcal{NP}$-complete by giving a reduction from the following problem (which is know to be $\mathcal{NP}$-complete): given a graph $G = (V, E)$, and a special vertex $s \in V$, does the graph have a *Hamilton path* (i.e. a path that visits every vertex exactly once), starting at s?

(c) [2 pts] What is the significance of (a) and (b) for the question of whether there is a fast algorithm to solve the "Please Take Me Out Of Here" problem (and the "Find Sophie" Facebook puzzle)?

Problem 8 Bonus Points [bonus_points] Regardless of what you write here (if anything), the grade you get for this problem equals the bonus points you have earned (if any), throughout the course, by solving exercise sets & labs. Feel free to express yourself artistically here if you like.