DAA 2007: Final Exam

[Answer questions worth 60 marks from the following. **Maximum marks: 60**]

Q1.[5] We are given a weighted directed graph $G = (V, E)$ in which edges that leave the source vertex s may have negative weights, all other edge weights are non-negative, and there are no negative weight cycles. Does Dijkstra's algorithm correctly finds shortest paths from s in this graph? Why/why not?

Q2.[5] Consider the following optimization version of the satisfiability problem: rather than decide whether there is an assignment that satisfies *all* the clauses, we instead seek an assignment that *maximizes* the number of satisfied clauses. This problem, known as MAX-SAT, is NP-hard. Given a set of m clauses in conjunctive normal form in n variables, give a simple randomized algorithm for MAX-SAT that computes a truth assignment to the variables such that the expected number of clauses satisfied by this assignment is at least $m/2$.

Q3.[5] Consider a data structure supporting deletes, find-mins and inserts on a set of integers where a delete(x) operation should delete x and all items whose value is at least x and an insert operation always inserts a value larger than what is already there. What data structure will you keep for this? What is the amortized time taken per deletion/find-min over a sequence of n delete operations starting with a data structure having n inserts?

Q4.[5] Give an $O(n^2)$ algorithm to find the longest monotonically increasing subsequence of a sequence of n numbers. (Example: given the sequence 4,8,5,7,2,9,10,1 your algorithm should output 4,5,7,9,10.)

Q5.[4] Let T be a minimum spanning tree of a graph G, and let L be the sorted list of the edge weights of T. Show that for any other minimum spanning tree T' of G, the list L is also the sorted list of edge weights of T'.

Q6.[4] We are given a number n and we are told that $n = p \cdot q$, where p and q are unknown primes. Suppose we are also given the size of the set $S = \{k : 1 \le k \le n \text{ and } \gcd(k, n) = 1\}$. How would you determine p and q efficiently?

Q7.[4] Recall Hall's Theorem. Show the following generalization: let d be a non-negative integer and $G = (A \cup B, E)$ an undirected bipartite graph such that for all $S \subseteq A, |N(S)| \ge |S| - d$. Then there is a matching of cardinality $|A| - d$ in G.

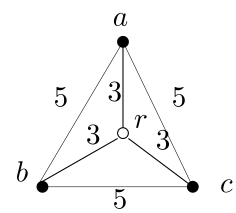
Q8.[4] Give a polynomial time algorithm to determine if a given graph $G = (V, E)$ admits a legal colouring of its vertices using just 2 colours, i.e., is there a function $f : V \to \{0, 1\}$ such that for each $(u, v) \in E$ we have $f(u) \ne f(v)$.

Q9.[7] This problem deals with an efficient technique for *verifying* matrix multiplication. The fastest known algorithm for multiplying two $n \times n$ matrices runs in $O(n^{2.376})$ time, which is significantly faster than the obvious $O(n^3)$ algorithm but this $O(n^{2.376})$ algorithm has the disadvantage of being extremely complicated. Suppose we are given an implementation of this algorithm and would like to verify its correctness. Since program verification is a difficult task, a reasonable goal might be to verify the correctness of the output produced on specific executions of the algorithm. In other words, given $n \times n$ matrices A, B, and C with entries from rational numbers, we would like to verify that $AB = C$. Note that here we want to use the fact that we do not have to compute C; rather, our task is to verify that the product is indeed C. Give an $O(n^2)$ time randomized algorithm for this problem with error probability at most 1/2.

[Hint: Choose a random vector $r \in \{0, 1\}^n$ (each entry in r is chosen independently and uniformly at random from $\{0, 1\}$) and proceed.]

Q10.[7] We are given an undirected complete graph $G = (V, E)$ with non-negative edge costs satisfying triangle inequality. The vertex set V is partitioned into two sets: *required* and *optional*. The problem of finding the minimum cost tree in G that contains all the required vertices and any subset of the optional vertices is NP-hard.

For example, in the graph below the vertices $\{a, b, c\}$ are the required vertices and r is an optional vertex. The tree defined by the edges $(r, a), (r, b), (r, c)$ is the minimum cost tree containing all the required vertices and its cost is 9.



Show that if one computes a minimum spanning tree on the required vertices, then the cost of this tree is always at most twice the optimal solution.

Q11.[8] The following problem is called the *Bin Packing* problem: Given n items with sizes $a_1, \ldots, a_n \in (0, 1]$, find a packing of all these items in unit-sized bins that minimizes the number of bins used.

(a) Show a factor 2 approximation algorithm for this problem.

(b) Assuming $P \neq NP$, show that for any $\epsilon > 0$, there is no approximation algorithm having a guarantee of $3/2 - \epsilon$ for the bin packing problem.

[Hint: Show that if there was a $(3/2 - \epsilon)$ factor algorithm, then the subset-sum problem can be solved in polynomial time. The subset-sum problem takes as input a set S of positive integers and a target integer t and has to say "yes" if there is a subset of S' of S whose elements sum to exactly t.]

Q12.[8] We want to relate two problems well-studied in computational geometry: the *half-plane intersection problem* and the *convex hull problem*. The convex hull problem takes as input n points $P = \{p_1, \ldots, p_n\}$ in the plane and computes the smallest convex set containing these n points. Let us assume that the points in P are such that the origin lies in the interior of the convex hull of P. We can also assume that no 3 points of P lie on the same straight line. The half-plane intersection problem takes as input n half-planes $\{h_1, h_2, \ldots, h_n\}$ where each h_i is given by $a_i x + b_i y \leq 1$ and computes the boundary enclosing the region $h_1 \cap h_2 \cap \cdots \cap h_n$.

For any point $q = (\alpha, \beta)$ (other than the origin), define the corresponding line ℓ_q as $\alpha x + \beta y = 1$ and the corresponding plane h_q as $\alpha x + \beta y \leq 1$. We will relate the half-plane intersection problem and the convex problem by comparing the boundary of the convex hull of $\{p_1, \ldots, p_n\}$ with the boundary of the intersection of half-planes $\{h_{p_1}, \ldots, h_{p_n}\}$.

Show that

(a) if p_i lies on the boundary of the convex hull of $\{p_1, \ldots, p_n\}$, then ℓ_{p_i} has to occur on the boundary of the region $h_{p_1} \cap h_{p_2} \cdots \cap h_{p_n}$.

(b) if $\overline{p_i p_j}$ is an edge of the convex hull, then the point $\ell_{p_i} \cap \ell_{p_j}$ is a vertex of the boundary of the region $h_{p_1} \cap h_{p_2} \cdots \cap h_{p_n}$.

Q13.[10] Show an $O(m\sqrt{n})$ algorithm to compute a maximum cardinality matching in a bipartite graph with m edges and n vertices.