Assignment V for AI2615 (Spring 2022)

May 17, 2022

Due: Sunday, May 29, 2022.

Problem 1 (40 points)

Recall the algorithm to find the k-th smallest element in an array with a random pivot. The running time of the algorithm is a random variable, which depends on the pivot we chose in each iteration. Given as input an array of length n, we also learnt that the algorithm uses $\Omega(n^2)$ steps to terminate in the worst case while on average it costs $O(n)$. In this homework, we develop an important tool to establish lower bounds for the expected running time of a *best* randomized algorithms for a problem.

Note that *deterministic* algorithms are special cases of randomized algorithms.

Let us fix a computational problem Π (e.g. sorting, finding the k-th smallest element, etc). Let $\mathcal{X}$ be the set of inputs of length n for some fixed n and let $\mathcal{A}$ be the set of *deterministic* algorithms solving Π. We assume for convenience that both $\mathcal{X}$ and $\mathcal{A}$ are finite.

(1) Now suppose we are given a randomized algorithm $\overline{A}$ (like the random pivoting algorithm just mentioned, $\overline{A}$ terminates in finite steps while the termination time is random). Prove that we can find a distribution $\mathscr{A}$ over $\mathcal{A}$ so that the expected running time of $\overline{A}$ on the input x equals

$$\mathbf{E}_{A\sim\mathscr{A}}[T(A,x)],$$

where $T(A,x)$ is the running time of A on the input x. Use this to conclude that the expected running time of the best algorithm for Π (over inputs with length n) is

$$\min_{\substack{\texttt{distribution} \\ \mathscr{A}\ \texttt{over}\ \mathcal{A}}} \max_{x\in\mathcal{X}} \mathbf{E}_{A\sim\mathscr{A}}[T(A,x)].$$

The expected running time of the best algorithm for Π is

$$\min_{\substack{\texttt{randomized algorithm} \\ \overline{A}\ \texttt{solving}\ \Pi}} \max_{x\in\mathcal{X}} \mathbf{E}\left[T(\overline{A},x)\right],$$

where the randomness is over those in the execution of $\overline{A}$.

(2) Use Von Neumann's minimax theorem to prove that

$$\max_{\substack{\texttt{distribution} \\ \mathscr{X}\ \texttt{over}\ \mathcal{X}}} \min_{a\in\mathcal{A}} \mathbf{E}_{X\sim\mathscr{X}}[T(a,X)] = \min_{\substack{\texttt{distribution} \\ \mathscr{A}\ \texttt{over}\ \mathcal{A}}} \max_{x\in\mathcal{X}} \mathbf{E}_{A\sim\mathscr{A}}[T(A,x)].$$

The LHS is the "expected time of the best deterministic algorithm against the worst input distribution".

This is the famous Yao's minimax principle, which essentially says that in order to lower bound the expected running time of the best algorithm for Π, one only needs to construct a distribution over all inputs (of length n) and show that every deterministic algorithm performs bad on average when the inputs are drawn from the distribution.

(3) Given an array A of n numbers and a number x which is in the array. Your task is to determine the location of x, i.e., to find the index $i \in [n]$ such that $A[i] = x$. Assume probing the value of the array at a specific location costs 1 unit of time (that is, reading the value of $A[j]$ for any j costs 1 unit of time). What is the worst running time for a deterministic algorithm? How to improve it using randomization? What's the expected cost of your randomized algorithm?

(4) Prove that each randomized algorithm for the above problem costs at least $\frac{n+1}{2} - \frac{1}{n}$ in expectation in the worst case. Does your algorithm match this lower bound?

Problem 2 (30 points)

Let $G = (V_1 \cup V_2, E)$ be a bipartite graph where $E \subseteq V_1 \times V_2$. A *perfect matching* of G is a set of edges $M \subseteq E$ touching each vertex in $V_1 \cup V_2$ *exactly once*. That is, for every $v \in V_1 \cup V_2$, there exists exactly one edge $e \in M$ such that $v \in e$. Consider the problem of determining whether a given graph contains a perfect matching.

(1) Show how to find the perfect matching (if any) by reducing to the problem of maxflow.

(2) *Hall's condition* is a combinatorial characterization of the existence of perfect matching:

Proposition 1 *The graph G contains a perfect matching* if and only if *for any $S \subseteq V_1$, $|N(S)| \geq |S|$, where $N(S) = \{u \in V_2 \mid \{u,v\} \in E \text{ for some } v \in S\}$ is the set of neighbours of vertices in S.*

Hint: Maxflow-Mincut Theorem.

Prove above proposition.

Problem 3 (30 points)

You and your $n-1$ roommates are always sharing expenses (bills, groceries, pizza, etc.) but it's terribly inconvenient to split each bill equally. You agree that each bill should be paid by one person (a possibly different one for each bill) who writes down what a subset of the roommates owe him. At the end of the year you aggregate everything in a *debt network* $G = (V, E, w)$ where $V = \{1, \ldots, n\}$ and $w(u, v)$ is the net (positive) amount that u owes v. If u owes v nothing then $(u, v) \notin E$. Prove that all debts can be settled with at most $n-1$ person-to-person payments *such that if u pays v then $(u, v) \in E$*.

Hint: Prove that the debt network can be equivalently transform to a network with at most $n-1$ edges, using the idea behind Ford-Fulkerson.

Problem 4

How long does it take you to finish the assignment (including thinking and discussion)?

Give a rating (1,2,3,4,5) to the difficulty (the higher the more difficult) for each problem.

Do you have any collaborators? Please write down their names here.