

Lecture 4

September 10, 2019

Scribes: Prof. Eric Price

NOTE: THESE NOTES HAVE NOT BEEN EDITED OR CHECKED FOR CORRECTNESS

Ying-hao Chen, Atharva Pendse

1 Explanation of HW 2 Problem 1

- Decision problems (is min-cut $\leq k$?) v.s. Optimization problems (what is min-cut?)
- If the answer to optimization problem $\in [B]$ and decision problem takes $T(n)$ time for $\frac{3}{4}$ success probability,
 - \Rightarrow binary search gives optimization in time $T(n) \log B$ with $\frac{3}{4}$ success probability.
- If ans to optimization problem $\in [B]$ and decision problem takes $T(n)$ time for $1 - \frac{1}{4 \log B}$ success probability,
 - $\Rightarrow \mathbb{P}[\text{optimization problem fails}] \leq \sum \mathbb{P}[\text{decision problem fails}]$
 $\leq \log B \cdot \max \mathbb{P}[\text{decision problem fails}]$
- By amplification, we can solve optimization problem with $\frac{3}{4}$ prob. in $T \log B \log \log B$ time. That means, at each level we repeat the decision problem $O(\log \log B)$ times to amplify the probability of getting a correct answer. Proof can be obtained using union bound.
- To construct a $T \log B$ time algorithm, one can consider how to recover from getting to the wrong branch in binary search (go back to the parent node when something happens).

2 Fun Probability Exercise

Q1: Given a standard die, roll it until we see a 6. How many rolls do we need in expectation?

1. Consider it as a geometry distribution with $p = \frac{1}{6}$.

$$\mathbb{E}[\#\text{rolls}] = \sum_{i=1}^{\infty} \left(\frac{5}{6}\right)^{i-1} \frac{1}{6} \cdot i = 6$$

2. Define a recursive relationship. Let $X = \mathbb{E}[\#\text{rolls}]$.

$$X = \frac{5}{6} \cdot (X + 1) + \frac{1}{6} \cdot 1$$

By solving this equation, we can get $X = 6$.

Q2: As the previous problem, what is $\mathbb{E}[\#\text{rolls until find 6} \mid \text{all outcomes are even}]$? In other words, we will start over with the process if we ever roll an odd number.

1. Suppose that we throw away all the runs with odd outcomes. The intuition is that as the run becomes longer, it's more likely to be thrown away. Hence, it's biased toward short sequence of rolls. We know that we will stop as soon as we see 1, 3, 5, and 6. Using the same logic as the previous problem, we can use property geometry distribution with $p = \frac{4}{6}$ to show that

$$\mathbb{E}[\#\text{rolls}] = \frac{1}{p} = \frac{6}{4} = 1.5$$

Note that the distribution of $\#\text{rolls ending at 1, 3, 5, 6}$ are the same (They are symmetric). Let R be the number of rolls until stops. Then

$$\mathbb{E}[\#\text{rolls until we meet 6} \mid \text{all outcomes are even}] = \mathbb{E}[R \mid \text{find roll in 6}] = \mathbb{E}[R] = 1.5$$

2. Could we have defined a recursive relationship like before? Using that method, we have

$$X = \frac{3}{6}X + \frac{2}{6}(X + 1) + \frac{1}{6} \cdot 1$$

That is, we have $\frac{3}{6}$ probability to reset (if we roll an odd number), $\frac{2}{6}$ probability to proceed (if we roll a 2 or 4), and $\frac{1}{6}$ probability to stop (if we roll a 6). The answer is no, this method won't work. The problem is that in the second term, it should be $(X + \mathbb{P}[\text{this roll count}])$ instead of $(X + 1)$, since there is still a non-zero probability to be reset. However, what is $\mathbb{P}[\text{this roll counts}]$? Actually,

$$\mathbb{P}[\text{this roll counts}] = \mathbb{P}[\text{we never see a 1,3, or 5}] = \sum_{i=1}^{\infty} \left(\frac{2}{6}\right)^{i-1} \frac{1}{6} = \frac{1/6}{1 - 2/6} = \frac{1}{4}$$

Hence, we can define $X = \mathbb{E}[\#\text{rolls}]$ and

$$X = \frac{3}{6}X + \frac{2}{6} \left(X + \frac{1}{4}\right) + \frac{1}{6} \cdot 1$$

We can now solve for X, giving us $X = 1.5$ as expected.

3 Von Neumann's Theorem

Suppose that Alice and Bob are playing a game where there's a payoff matrix V . Alice can choose a row of V and Bob can choose a column of V . Alice wants to minimize the payoff but Bob wants to maximize it. This is a zero-sum game. Suppose that Alice can use a mixed strategy with distribution p over the rows and Bob can use a mixed strategy with distribution q over the columns. The expected payoff can be described as

$$p^T V q$$

Actually, according to Von Neumann's theorem,

$$\min_p \max_{e_i} p^T V e_i = \min_p \max_q p^T V q = \max_q \min_p p^T V q = \max_q \min_{e_j} e_j^T V q$$

No matter who goes first, the expected payoff are the same. Also, the second people can always choose his(her) best strategy deterministically.

3.1 Recall

- Deterministic alg. needs $\Omega(n)$ queries in game tree evaluation.
- Non-deterministic alg. needs $\Theta(\sqrt{n})$ queries in game tree evaluation.

Suppose we put all the run time information to a cost matrix V , where each column represents a game tree instance and each row represents a deterministic algorithm. Note that there's at least one n on each row.

$$\min_{\text{alg } \mathcal{A}} \max_{\text{instance } \mathcal{I}} V(\mathcal{A}, \mathcal{I}) = n$$

There's some $\Theta(\sqrt{n})$ on each column.

$$\max_{\text{instance } \mathcal{I}} \min_{\text{alg } \mathcal{A}} V(\mathcal{A}, \mathcal{I}) = \Theta(\sqrt{n})$$

According to previous lecture, we know that the expected number of queries for randomized algorithm is bounded by

$$\min_{\substack{p:\text{distr.} \\ \text{over algs}}} \max_{\text{instance } \mathcal{I}} p^T V e_{\mathcal{I}} \leq n^{0.793}$$

By Von Neumann's theorem, we have

$$\max_{\substack{q:\text{distr.} \\ \text{over instances}}} \min_{\mathcal{A}:\text{det. alg}} e_{\mathcal{A}}^T V q = \min_{\substack{p:\text{distr.} \\ \text{over algs}}} \max_{\text{instance } \mathcal{I}} p^T V e_{\mathcal{I}} \leq n^{0.793}$$

This is known as "Yao's principle." Suppose we want to know the performance of the optimal randomized algorithm, when performed on its worst-case input for that algorithm. Yao's principle says that it suffices to consider the worst-case fixed *distribution* on inputs, and analyze the performance of the best deterministic algorithm for that input.

That is to say, suppose we can show for a particular distribution on inputs that no deterministic algorithm can perform better than X . Then any randomized algorithm, being just a mixture of deterministic algorithms, also performs no better than X . Yao's principle says that this is without loss of generality: there exists such a distribution on inputs so that the lower bound for randomized algorithms derived from this approach is optimal.

Goal: pick some distr. q s.t. no det. alg. does well on it. Consider a NAND tree where every node is active with probability p . In this case,

$$1 - p = \mathbb{P}[\text{not active}] = \mathbb{P}[\text{both child active}] = p^2$$

By solving $p^2 + p - 1 = 0$, we'll have $p = \frac{\sqrt{5}-1}{2}$. In this distribution, the expected number of leafs that our randomized game tree algorithm from previous lecture needs to explore is

$$\begin{aligned} T(h) &\leq T(h-1)(1-p) + 2T(h-1)p \\ &= T(h-1)(1+p) \\ &\leq (1+p)^h \\ &= (2^{\log_2(1+p)})^h \\ &= (2^h)^{\log_2(1+p)} \\ &= n^{\log_2(1+p)} \leq n^{0.6943} \end{aligned}$$