CS3230 Semester 2 2024/2025

Design and Analysis of Algorithms

Tutorial 06
Randomized Algorithms
For Week 07

Document is last modified on: February 18, 2025

MODEL ANSWER IS FOR OUR CLASS ONLY; NOT TO BE DISTRIBUTED IN PUBLIC

1 Lecture Review: Randomized Algorithms

Techniques: Linearity of expectations, indicator random variables, Markov inequality, union bound, principle of deferred decision, amplification of success probability.

Algorithms: Freivalds' algorithm (Monte Carlo), (Randomized) Quick Sort (Las Vegas).

Balls and bins: coupon collector (probability of no empty bin), chain hashing (expected bin size).

2 Tutorial 06 Questions

Q1). In the class, we showed that Freivalds' algorithm succeeds with a probability of at least $1/2$. Show that the bound $1/2$ in the analysis is actually the best possible by constructing an input (A, B, C) on which the success probability of Freivalds' algorithm is precisely $1/2$.

Consider 1×1 matrices $A = (1)$, $B = (0)$, and $C = (1)$. We have $AB \neq C$. In Freivalds' algorithm, $v = (v_1)$, where v_1 is chosen from $\{0, 1\}$ uniformly at random. With probability $1/2$, $v_1 = 0$, in which case[1] $A(Bv) = Cv$ (the answer is incorrect). With probability $1/2$, $v_1 = 1$, in which case $A(Bv) \neq Cv$

[1]Notice that when computing ABv, we need to compute matrix-row vector (Bv) of $A(Bv)$ first to avoid triggering the actual complex matrix multiplication of AB

(the answer is correct). By appending zeros, the construction can be extended to $n \times n$ matrices for every positive integer n. For example, if $n = 3$, then we can use

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \text{and} \quad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

and we still have ($A(Bv) = Cv$ if $v_1 = 0$) and ($A(Bv) \neq Cv$ if $v_1 = 1$).

For Q2). and Q3). Consider the equality testing problem, where Alice holds an n-bit string $S_A \in \{0,1\}^n$, Bob holds an n-bit string $S_B \in \{0,1\}^n$, and they want to decide whether $S_A = S_B$. Consider the following communication protocol, where an n-bit string is seen as a number expressed in base-2.

1. Let S be the set of n^2 smallest prime numbers.
2. Alice samples a number p from S uniformly at random.
3. Alice sends p and $S_A \bmod p$ to Bob (thus, only $O(\log p) \subseteq O(\log n)$ bits are sent, see Q3).
4. After receiving Alice's message, Bob calculates $S_B \bmod p$.
5. If $S_A \bmod p = S_B \bmod p$, Bob decides that $S_A = S_B$, otherwise Bob decides that $S_A \neq S_B$.

Q2). Show that this (randomized) communication protocol is correct with a probability of at least $1 - \frac{1}{n}$.

We consider the 2 cases:

- **Case $S_A = S_B$:** Bob always decides that $S_A = S_B$ because $S_A \bmod p = S_B \bmod p$ for all p, so the output is correct with probability 1.
- **Case $S_A \neq S_B$:** the output is incorrect only if $S_A \bmod p = S_B \bmod p$, that is, p is a divisor of $|S_A - S_B|$. Since $|S_A - S_B| \leq 2^n$, $|S_A - S_B|$ has at most n prime factors, so the probability that p is a divisor of $|S_A - S_B|$ is at most $n/|S| \leq 1/n$.

Thus, the algorithm is correct with probability at least $1 - \frac{1}{n}$.

Remark: By the prime number theorem, $p \in O(n^2 \log n)$, so this communication protocol only requires communicating $O(\log p) \subseteq O(\log n)$ bits. This demonstrates an exponential separation between randomized and deterministic algorithms, as any deterministic algorithm solving the equality testing problem requires communicating $\Omega(n)$ bits in the worst case.

Q3). Show that any deterministic algorithm solving the equality testing problem requires communicating $\Omega(n)$ bits in the worst case.

For the special case of one-way communication (Alice sends a message to Bob, and then Bob decides the answer), a lower bound of n bits can be seen as follows.

- The number of possible n-bit strings S_A is 2^n.

- The number of possible $(n-1)$-bit messages sent by Alice is 2^{n-1}.

By the pigeonhole principle, there must be two n-bit strings X and Y such that the $(n-1)$-bit message that Alice sends to Bob when $S_A = X$ is identical to the $(n-1)$-bit message that Alice sends to Bob when $S_A = Y$, meaning that Bob cannot distinguish between the two cases $S_A = X$ and $S_A = Y$, so the algorithm must fail in one of the two cases: $(S_A = X, S_B = X)$ and $(S_A = Y, S_B = X)$.
Remarks: For the more general case where Alice and Bob can communicate with each other in multiple rounds in both directions, there is still a lower bound of n bits, see the following Wikipedia page: `https://en.wikipedia.org/wiki/Communication_complexity`.

For Q4). and Q5). You are given a graph $G = (V, E)$ (without self-loops) and your task is to partition its vertex set into two parts $V = V_1 \cup V_2$ randomly as follows.

- Each vertex $v \in V$ flip an unbiased coin independently.
  - If the outcome is head, which happens with probability $1/2$, v joins V_1.
  - If the outcome is tail, which happens with probability $1/2$, v joins V_2.

Q4). Show that the expected number of edges crossing V_1 and V_2 is exactly $|E|/2$.

For each edge $e \in E$, let X_e be the indicator[2] random variable for the event that e crosses V_1 and V_2, we first compute $\mathbf{E}[X_e]$ for any edge $e = \{u, v\}$:

- With probability $1/4$, $u \in V_1$ and $v \in V_1$, in which case e does not cross V_1 and V_2.
- With probability $1/4$, $u \in V_1$ and $v \in V_2$, in which case e crosses V_1 and V_2.
- With probability $1/4$, $u \in V_2$ and $v \in V_1$, in which case e crosses V_2 and V_1 (which also crosses V_1 and V_2 as edge orientation doesn't matter).
- With probability $1/4$, $u \in V_2$ and $v \in V_2$, in which case e does not cross V_1 and V_2.

Therefore, indeed $\mathbf{E}[X_e] = 1 \cdot \mathbf{Pr}[e \text{ crosses } V_1 \text{ and } V_2] = 1/2$. Now considering the expected number of edges crossing V_1 and V_2:

$$
\begin{aligned}
\mathbf{E}[X] &= \mathbf{E}[\sum_{e \in E} X_e] && (X = \sum_{e \in E} X_e) \\
&= \sum_{e \in E} \mathbf{E}[X_e] && \text{(Linearity of expectation)} \\
&= \sum_{e \in E} \frac{1}{2} && \text{(Each edge } \mathbf{E}[X_e] = 1/2) \\
&= \frac{|E|}{2}. && \text{(Sum over all edges)}
\end{aligned}
$$

[2] Takes the value 1 if a specific event occurs and 0 otherwise.

Remark: As a corollary, we obtain the following result in graph theory.

- Any graph $G = (V, E)$ admits a cut of size of at least $|E|/2$.

Here a cut is a partition of the vertex set into two parts, and the size of a cut is the number of edges crossing the two parts.

Q5). Is it possible to improve the bound $|E|/2$ in the above result?
If you claim it is possible, propose some ideas and analyze the new bound.

Assuming $E \neq \emptyset$, a slight improvement can be achieved in many different ways by modifying the random partitioning algorithm. For example, we can pick one edge $e^* = \{u^*, v^*\}$ and force $u^* \in V_1$ and $v^* \in V_2$. The remaining vertices $V \setminus \{u^*, v^*\}$ are still assigned to V_1 or V_2 randomly. Observe that

$$\mathbf{E}[X_e] = \begin{cases} 1 & \text{if } e = e^*, \\ \frac{1}{2} & \text{if } e \in E \setminus \{e^*\}, \end{cases}$$

so $\mathbf{E}[X] = \sum_{e \in E} \mathbf{E}[X_e] = 1 + \frac{|E|-1}{2} = \frac{|E|+1}{2}$, which is slightly better than $\frac{|E|}{2}$.

If $|V|$ is an even number, a bound of $\frac{|E|}{2} \cdot \frac{|V|}{|V|-1}$ can be attained by selecting V_1 uniformly at random from the collection of all $(|V|/2)$-vertex subsets of V and setting $V_2 = V \setminus V_1$, in which case $\mathbf{E}[X_e] = \frac{|V|/2}{|V|-1}$, so $\mathbf{E}[X] = \sum_{e \in E} \mathbf{E}[X_e] = \frac{|E|}{2} \cdot \frac{|V|}{|V|-1}$. **Remarks:** This bound is tight for complete graphs, and a bound that is tight for complete graphs can also be obtained similarly for odd $|V|$.