MAX PLANCK INSTITUTE
FOR INFORMATICS

SIC Saarland Informatics Campus

Lecturers: **Evangelos Kipouridis and Tomasz Kociumaka** **Winter 2025/26**
Tutors: **Anouk Duyster and Yonggang Jiang**

# Randomized and Approximation Algorithms, Exercise Sheet 1

`https://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter-2025/26/randomized-and-approximation-algorithms`

Total Points: **100** + 35 bonus points Due: **14:00** on Thursday, **October 30**, 2025

## Exercise 1 — **6 + 6 + 6 + 6 + 6** points

Let $X, Y, Z$ be subsets of an $n$-element universe, sampled independently and uniformly at random. Compute the following quantities. Justify each answer.

**a** $\mathbb{P}[X \subseteq Y]$;

**b** $\mathbb{E}[|X \setminus Y|]$;

**c** $\mathbb{P}[X \subseteq Y \mid X \subseteq Z]$;

**d** $\mathbb{E}[|X \cap Y \cap Z| \mid X \cup Y = Z]$;

**e** $\mathbb{E}[|X| \mid X \subseteq Y]$.

*Hint:* There are two equivalent definitions of a uniformly random subset $X$ of a universe $U$:

- For each subset $V \subseteq U$, we have $\mathbb{P}[X = V] = 2^{-|U|}$.
- For each element $u \in U$, we have $\mathbb{P}[u \in X] = \frac{1}{2}$ and the events $\{u \in X\}$ are mutually independent across $u \in U$.

## Exercise 2 — **10** + **15** points + 15 bonus points

Suppose that Alice and Bob have length-$n$ binary strings $A, B \in \{0,1\}^n$. During the lecture, we discussed a one-way communication protocol in which Alice sends an $\mathcal{O}(\log n)$-bit message to Bob so that he:

- always answers '*equal*' if $A = B$;
- answers '*equal*' with probability at most $\frac{1}{n}$ if $A \neq B$.

Recap: Alice picks the smallest prime $p \in \{n^2, \dots, 2n^2\}$ (it exists by Bertrand–Chebyshev theorem), draws $x \in \{0, \dots, p-1\}$ uniformly at random, computes $\Phi_x(A) = (\sum_{i=0}^{n-1} A[i] \cdot x^i) \bmod p$, and sends a tuple $\langle x, \Phi_x(A) \rangle$. Having received the message, Bob evaluates $\Phi_x(B) := (\sum_{i=0}^{n-1} B[i] \cdot x^i) \bmod p$ and answers '*equal*' if and only if $\Phi_x(B) = \Phi_x(A)$. The message size is $2\lceil \log p \rceil \leq 2\log(2p) \leq 2\log(4n^2) = 4\log(2n)$.

**a** Modify the protocol (keeping the guarantees $\frac{1}{n}$ on the error probability and $4\log(2n)$ on the message size) so that Alice's only source of randomness is a single uniformly random string $R \in \{0,1\}^{\lceil 2 \log n \rceil}$. As we discussed, this does not allow obtaining a uniformly random $x \in \{0, \dots, p-1\}$ (unless $p = 2$).

In the remainder of this exercise, Alice and Bob are given a **shared** uniformly random string $R \in \{0,1\}^m$.

**b** Prove that, if $m \geq n$, then Alice can send a 1-bit message so that Bob:

- always answers '*equal*' if $A = B$;
- answers '*equal*' with probability at most $\frac{1}{2}$ if $A \neq B$.

**c** Prove that, if $m \geq 10 \cdot \log(2n)$, then Alice can send a 1-bit message so that Bob:

- always answers '*equal*' if $A = B$;
- answers '*equal*' with probability at most $\frac{2}{3}$ if $A \neq B$.

*Hint:* Combine the ideas from the points above.

## Exercise 3 — 5 + 15 + 20 + 5 points + 20 bonus points

Let $\mathrm{MinCut}(G)$ denote the size of the global minimum cut of a multigraph $G$. Recall the algorithm from the lecture:

**Algorithm 1:** The min-cut algorithm from the lecture.

```
function Contract($G, t$):
    while $|V(G)| > t$ do
        Sample $e \in E(G)$ uniformly at random;
        $G \leftarrow G/e$;                                    // Contract edge $e$
    return $G$;

function SimpleMinCut($G$):
    $G \leftarrow$ Contract($G, 2$);
    return $|E(G)|$;
```

**a** Prove that, for every multigraph $G$ satisfying $|V(G)| \geq t$, we have

$$\mathbb{P}[\mathrm{MinCut}(G) = \mathrm{MinCut}(\texttt{Contract}(G,t))] \geq \frac{t(t-1)}{n(n-1)} > \frac{(t-1)^2}{(n-1)^2}.$$

**b** Describe how to implement $\texttt{Contract}(G,t)$ in $\mathcal{O}(|V(G)|^2)$ time when the multigraphs are represented by adjacency matrices. Assume that a uniformly random integer from any range of size at most $|E(G)|$ can be generated in $\mathcal{O}(1)$ time.

*Hint:* This is much easier than an $\widetilde{\mathcal{O}}(|V(G)| + |E(G)|)$-time implementation using adjacency lists.

In the remainder of this exercise, you will analyze algorithms based on the following scheme, where $t(n)$ and $r(n)$ are parameters to be chosen; you can recover `SimpleMinCut` using $t(n) = 2$ and $r(n) = 1$.

**Algorithm 2:** A recursive global min-cut algorithm.

```
function RecursiveMinCut($G$):
    $n \leftarrow |V(G)|$;
    if $n \leq 2$ then return $|E(G)|$;
    $G \leftarrow$ Contract($G, t(n)$);
    $mincut \leftarrow \infty$;
    for $t \leftarrow 1$ to $r(n)$ do
        $mincut \leftarrow \min(mincut,$ RecursiveMinCut($G$)$)$;
    return $mincut$;
```

**c** Consider $t(n) = \lceil \frac{n+1}{2} \rceil$ and $r(n) = 5$. Prove that, for every multigraph $G$ with $n \geq 2$ vertices:

- $\texttt{RecursiveMinCut}(G)$ works in $\mathcal{O}(n^{\log_2 5}) \leq \mathcal{O}(n^{2.322})$ time, and
  *Hint:* Calculations are cleaner if you express the time complexity in terms of $2^{\lceil \log(n-1) \rceil}$ instead of $n$.
- $\texttt{RecursiveMinCut}(G)$ is correct with probability at least $\frac{1}{10}$.
  *Hint:* Use induction on $n$.

**d** Propose values for $t(n)$ and $r(n)$ such that, for every multigraph $G$ with $n \geq 2$ vertices:

- $\texttt{RecursiveMinCut}(G)$ works in $\mathcal{O}(n^{2.01})$ time, and
- $\texttt{RecursiveMinCut}(G)$ is correct with probability $\Omega(1)$.

*Hint:* You can choose the parameters so that the calculations are analogous to part **c**.

**e** Consider $t(n) = \lceil \frac{n+1}{2} \rceil$ and $r(n) = 4$. Prove that, for every multigraph $G$ with $n \geq 2$ vertices:

- $\texttt{RecursiveMinCut}(G)$ works in $\mathcal{O}(n^2 \log n)$ time, and
- $\texttt{RecursiveMinCut}(G)$ is correct with probability $\Omega(1/\log n)$.

Argue that the global min-cut problem can be solved in $\mathcal{O}(n^2 \log^2 n)$ correctly with probability $\geq \frac{2}{3}$.