3^{RD} TUTORIAL ON RANDOMIZED ALGORITHMS

Random walks into probabilistic complexity classes

1. *d-regular graphs.* Show that the cover time of d-regular undirected graphs is $O(n^2 \log n)$. Hint: Use bounds from the lecture. What is the diameter of d-regular graphs, i.e., the largest possible distance between two nodes?

Compare this bound with the bounds for complete graphs and lollipops from the last tutorial.

2. *Random Tom and Jerry on a cycle.* A cat and a mouse each independently take a random walk on a cycle of length $2n+1$. (In each step, they both take a random step, i.e., both random walks are synchronized.) The game ends (the cat eats the mouse) if they land at the same vertex at the same time. (If they traverse the same edge in the opposite directions, nothing happens.) Find the expected number of steps of the game if the cat and the mouse start (a) at adjacent nodes and (b) at distance n (i.e., at the largest possible distance).

3. *Classes of randomized algorithms I.* Show that if NP $\subseteq$ BPP then NP=RP.

4. *Simulating a biased coin using a fair coin.* We are given a fair coin with $\Pr[\text{tails}] = 0.5$. Show how to generate a random bit with $\Pr[1] = p$ for a given $p \in (0,1)$ (both $p = 0$ and $p = 1$ are a bit boring). How many fair coin flips do we need in expectation?

i) First, assume that $p = k/2^\ell$ for some integers k and ℓ.

ii) What about any rational p, say, $p = 1/3$? What about irrational p?

5. *Classes of randomized algorithms II.* Let PP be the class of languages L such that there is a probabilistic Turing machine A that satisfies the following:

- A runs in polynomial time (with probability 1).
- If $x \in L$ then $\Pr[A \text{ accepts } x] > 1/2$.
- If $x \notin L$ then $\Pr[A \text{ accepts } x] < 1/2$.

Show that NP $\subseteq$ PP $\subseteq$ PSPACE. What does it mean for the definition of BPP?

6. *Eigenvalues warm-up.* Let $A \in \mathbb{R}^{n\times n}$ be a matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$. Show that the matrix $A + dI_n$ has eigenvalues $d + \lambda_1, \ldots, d + \lambda_n$.