Lecture 19

Linearity of Expectation

1. Write a proof of linearity of expectation.

2. If you toss an unbiased coin $n$ times, what is the expected number of heads?

3. For a sequence on $n$ independent coin tosses, what is the expected number of two consecutive heads?

4. Suppose you put $m$ balls into $n$ bins randomly and independently, that is, for each ball the probability that it falls into a particular bin is $1/n$. What is the expected number of empty bins?

5. There are 48 students in our class. Suppose each student has a uniformly random birthday, i.e., the probability that any particular date is the student’s birthday is $1/365$. What is the expected number of pairs of students who have a common birthday.

6. Suppose you have a random number generator that gives an integer between 0 and 9, each with probability $1/10$. What is the expected number of trials till you see each of the ten numbers at least once.

Independence: Suppose you choose a random integer $x$ between 1 and 60. Which of the following events are dependent or independent of each other? First try to answer intuitively and then see what the calculations say.

- $x$ is divisible by 4.
- $x$ is divisible by 5.
- $x$ is divisible by 6.

Maximum Satisfiability

- In the lecture, we saw some randomized algorithms with certain approximation guarantees. We also discussed some ideas on how to convert them into a deterministic algorithm. Write those deterministic algorithms formally and prove that the same approximation guarantees hold for them.
• Complete the analysis of the $(3/4)$-approximation algorithm.

• Consider the Max SAT instance with four clauses $(x_1 \lor x_2) \land (x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2)$, and each clause having weight 1. Prove that the integrality gap for the LP we wrote is $3/4$. That is, the LP optimal value is 4 and the actual optimal value is 3.

Max Directed Cut: We want to find a maximum weight directed cut. That is, given a directed graph with edge weights $(w_{i,j}$ for the edge going from $i$ to $j$), we want to find a subset of vertices $U$ so that the sum $\sum_{i \in U, j \not\in U} w_{i,j}$ is maximized. Write an integer program for this problem.