

CS 671 (Fall 2020) — Assignment 3

Due: 10/14/2020

Read Chapters 3 and 4 from the textbook. Here are the homework problems:

- (1) With Halloween not too far away, we have to start worrying about zombie attacks. Here is how zombie attacks work. You start with just one zombie in a population of n people. In each round, each zombie independently picks some moving thing uniformly at random and bites it. If that moving thing was another zombie, nothing happens. If it was a human, he/she will now also turn into a zombie for the next round. Note that multiple zombies might choose to bite the same zombie or human.

We are interested in how many rounds it will take until there are no more humans left. Show that with high probability, this happens after no more than $O(\log n)$ rounds.

(Note: this problem is not exactly easy. You may want to think about how the number of zombies behaves during different phases. As in: you might need multiple different analysis techniques during the early rounds and the later rounds of the zombie attack.)

- (2) Suppose that we generate a string x of length n over an alphabet Σ (such as a genome over $\Sigma = \{a, c, g, t\}$) uniformly at random by generating each position independently and uniformly at random. We are now interested in how many occurrences of a pattern p there are in x . p is also a string (of length $k \ll n$) over Σ .

(a) Derive the expected number of occurrences of p in x .

(b) Show that the actual number of occurrences is sharply concentrated around the expectation. How sharp a bound can you obtain?

- (3) Here is a variant of the well-known HITTING SET problem. There is a universe U of n elements, and you have m possible sets $S_1, \dots, S_m \subseteq U$, with associated integer “coverage requirements” $r_1, \dots, r_m \geq 1$. To keep things clean, let’s assume that m is polynomial in n always.

You want to select a subset $T \subseteq U$ of elements, ensuring that you hit each set S_j at least r_j times, i.e., that $|T \cap S_j| \geq r_j$ for all j . Subject to that, you want to minimize the size of T .

Phrase this question as an integer linear program, and consider its fractional relaxation. Give a suitable rounding algorithm of this linear program, and prove that it gives an $O(\log n)$ approximation algorithm.