

We may grade a **subset of the assigned questions**, to be determined after the deadline, so that we can provide better feedback on the graded questions.

Unless otherwise stated, each question requires sufficient justification to convince the reader of the correctness of your answer.

For bonus questions, we will not provide any insight during office hours or Piazza, and we do not guarantee anything about the difficulty of these questions.

We strongly encourage you to typeset your solutions in L^AT_EX.

If you collaborated with someone, you must state their name(s). You must write your own solution for all problems and may not look at any other student's write-up.

0. If applicable, state the name(s) and username(s) of your collaborator(s).

Solution:

1. (a) A group of trees, all with distinct heights, have been planted uniformly randomly in a line. However, due to overcrowding, because some trees are taller than others, certain trees will struggle to receive sunlight. We will determine how many of the trees are expected to die due to this lack of sunlight. If any tree is shorter than both the trees directly in front and behind of it, then it does not receive enough sunlight to survive summer. The trees at the very front and back will die if they are shorter than their (only) neighbor. At the beginning of summer, there are n trees. Compute the expected number of trees that will die in one summer.

Solution:

- (b) A group of n students, all of whom have distinct heights, line up in a single-file line uniformly at random. If a student has any students in front of them who is taller than them, then they cannot see. For this reason, every student files one complaint for each taller student who is in front of them since each one of these students would individually block the original student from seeing.

For example, if there were 3 students, where the tallest student was in the front and the shortest student was in the back, there would be 2 complaints from the shortest student and 1 complaint from the middle student.

Compute the expected number of complaints.

Solution:

2. Let A be an array of length n of a random permutation of n distinct integers. Compute the expected number of increasing subarrays (i.e., contiguous subsequence) in A of length k , for $k \leq n$.

Solution:

3. In order to generate random bits, operating systems often use real-world events as a source of randomness (e.g., mouse movements, atmospheric noise, thermal fluctuations, even the radioactive decay of certain isotopes). Unfortunately, these events may not be uniformly and independently random, so there is often bias in the bits we generate from them. This presents a problem since most randomized algorithms rely on access to unbiased, uniformly independent random bits. We explore various ways to generate these desired unbiased results.

Hint: The geometric distribution gives the probability that the first occurrence of success requires k independent trials, each with success probability p . If the probability of success on each trial is p , then the probability that the k^{th} trial (out of k trials) is the first success is

$$\Pr [\text{The } k^{\text{th}} \text{ trial is the first success}] = p(1 - p)^{k-1}$$

Using this, we can find the expected number of trials until the *first* success,

$$\mathbb{E}[\text{number of trials until the first success}] = \sum_{k=1}^{\infty} kp(1 - p)^{k-1} = \frac{1}{p}$$

- (a) Let GET-BIASED-BIT be a function that returns 1 with probability p (where p is an **unknown** real number between 0 and 1 exclusive) and returns 0 with probability $1 - p$. Consider this randomized algorithm, GET-UNBIASED-BIT, that returns 1 or 0 with equal (nonzero) probability, using calls to GET-BIASED-BIT as a source of randomness.

```
1: function GET-UNBIASED-BIT()
2:   loop
3:      $b_1 \leftarrow \text{GET-BIASED-BIT}()$ 
4:      $b_2 \leftarrow \text{GET-BIASED-BIT}()$ 
5:     if  $b_1 = 0$  and  $b_2 = 1$  then
6:       return 0
7:     else if  $b_1 = 1$  and  $b_2 = 0$  then
8:       return 1
```

- i. Prove that the probability that this algorithm returns 0 is equal to the probability that it returns 1.

Solution:

- ii. How many calls does GET-UNBIASED-BIT make to GET-BIASED-BIT in expectation? Express your answer in terms of p .
(Recall the hint from the beginning of the problem.)

Solution:

- (b) Now let GET-BIASED-DIE be a function which simulates a biased 6-sided die by returning $\{1, 2, \dots, 6\}$ with an unknown nonuniform probability distribution $P = (p_1, p_2, \dots, p_6)$, where $\sum_i^6 p_i = 1$ and p_i is the probability that the biased die returns i .
- i. Write a randomized algorithm, GET-UNBIASED-DIE, that returns an element from $\{1, 2, \dots, 6\}$ uniformly at random (equal probability), using calls to GET-BIASED-DIE as a source of randomness.

Solution:

- ii. Prove that your provided algorithm outputs each integer in $\{1, 2, 3, 4, 5, 6\}$ with equal probability.

Solution:

- iii. How many calls does GET-UNBIASED-DIE make to GET-BIASED-DIE in expectation? Express your answer in terms of the probability distribution of the six outcomes (p_1, p_2, \dots, p_6) .
(Recall the hint from the beginning of the problem.)

Solution:

4. As its name suggests, the MAX-INDEPENDENT-SET problem concerns finding the maximal independent set in a given graph $G = (V, E)$. (Recall that an independent set S of G is a subset $S \subseteq V$, such that no two vertices share an edge. That is, it is a set of isolated vertices.) The following is a randomized $\frac{1}{\Delta+1}$ -approximation for the MAX-INDEPENDENT-SET problem (where Δ is the maximum degree of any vertex in the graph). Note that this is **not** a constant factor approximation since Δ depends on the graph.

```

1: function FIND-IND-SET( $G = (V, E)$ )
2:    $S \leftarrow \emptyset$ 
3:   for  $v \in V$  do
4:     Randomly sample a random real  $r_v \in [0, 1]$ 
5:   for  $v \in V$  do
6:     if  $r_v > r_{v'}$  for every neighbour  $v'$  of  $v$  then
7:        $S \leftarrow S \cup \{v\}$ 
8:   return  $S$ 

```

- (a) Prove that the algorithm returns an independent set S .

Solution:

- (b) Prove that it is a $(1/(\Delta + 1))$ -approximation *in expectation*. That is, if OPT is the size of the maximum independent set in G , prove that

$$\mathbb{E}[|S|] \geq \frac{\text{OPT}}{\Delta + 1},$$

meaning that the expected size of the output set S is within $(1/(\Delta + 1))$ of OPT.

Solution: