

Jerry Cain
CS 109

Problem Set #1
March 31, 2021

Problem Set #1 Due: 10:00am on Friday, April 9th

With problems by Chris Piech, Mehran Sahami and Lisa Yan

- Submit on Gradescope by 10:00am Pacific on Friday, April 9th, for a small, "on-time" bonus.
- All students have a pre-approved extension, or "grace period" that extends until Monday 10:00am Pacific, when they can submit with no penalty. **The grace period expires on 10:00am Pacific on Monday, April 12th**, after which we cannot accept further late submissions.
- **Collaboration policy:** You are encouraged to discuss problem-solving strategies with each other as well as the course staff, but you must write up your own solutions and submit individual work. Please cite any collaboration at the top of your submission.
- **Tagging written problems:** When you submit your written PDF on Gradescope you must tag your PDF, meaning that you must assign pages of your PDF as answers to particular questions so that we can properly grade your submission. For problem sets, we are deducting **2 points** for any submissions that do not have all questions tagged.
- **For each problem, briefly explain/justify how you obtained your answer.** Brief explanations of your answer are necessary to get full credit for a problem even if you have the correct numerical answer. The explanations help us determine your understanding of the problem whether or not you got the correct answer. Moreover, in the event of an incorrect answer, we can still try to give you partial credit based on the explanation you provide. It is fine for your answers to include summations, products, factorials, exponentials, or combinations; you don't need to calculate those all out to get a single numeric answer.
- If you handwrite your solutions, you are responsible for making sure that you can produce **clearly legible** scans of them for submission. You may also use any word processing software you like for writing up your solutions. On the CS109 webpage we provide a template file and tutorial for the \LaTeX system, if you'd like to use it.

1 Written problems

Submit your solutions to these written problems as a single pdf file on Gradescope.

1. How many ways can 10 people be seated in a row if
 - a. there are no restrictions on the seating arrangement?
 - b. persons A and B must sit next to each other?
 - c. there are 5 adults and 5 children, and no two adults nor two children can sit next to each other?
 - d. there are 5 married couples and each couple must sit together?

2. At the local zoo, a new exhibit consisting of 3 different species of birds and 3 different species of reptiles is to be formed from a pool of 8 bird species and 6 reptile species. How many exhibits are possible if
 - a. there are no additional restrictions on which species can be selected?
 - b. 2 particular bird species cannot be placed together (e.g., they have a predator-prey relationship)?
 - c. 1 particular bird species and 1 particular reptile species cannot be placed together?
3. Given all the start-up activity going on in high-tech, you realize that applying combinatorics to investment strategies might be an interesting idea to pursue. Say you have \$20 million that must be invested among 4 possible companies. Each investment must be in integral units of \$1 million. How many different investment strategies are available if
 - a. an investment must be made in each company, and we must invest all \$20 million? Assume we have a minimal investment requirement per company, where we must invest a minimal investment of \$1, \$2, \$3, and \$4 million dollars for company 1, 2, 3, and 4, respectively.
 - b. investments must be made in at least 3 of the 4 companies, and we must invest all \$20 million? Assume we have the same minimal investment as in part (a), where should we choose to invest in company n (for $n = 1, \dots, 4$), we must invest a minimal of $\$n$ million.
 - c. we must invest less than or equal to $\$k$ million dollars total among the 4 companies, where k is an integer such that $10 \leq k \leq 20$? Note that you can think of k as a constant that can be used in your answer. Assume in this part that we do not have a minimal investment.
4. If we assume that all possible poker hands (comprised of 5 cards from a standard 52 card deck) are equally likely, what is the probability of being dealt:
 - a. a flush? (A hand is said to be a flush if all 5 cards are of the same suit. Note that this definition means that *straight flushes* (five cards of the same suit in numeric sequence) are also considered flushes.)
 - b. two pairs? (This occurs when the cards have numeric values a, a, b, b, c , where a, b and c are all distinct.)
 - c. four of a kind? (This occurs when the cards have numeric values a, a, a, a, b , where a and b are distinct.)
5. Say we roll a six-sided die six times. What is the probability that
 - a. we will roll two different numbers *thrice* (three times) each?
 - b. we will roll *exactly one* number *exactly* three times? Hint: Be careful of overcounting.
6. A binary string containing M 0's and N 1's (in arbitrary order, where all orderings are equally likely) is sent over a network. What is the probability that the first r bits of the received message contain exactly k 1's?
7. Say we send out a total of 20 distinguishable emails to 12 distinct users, where each email we send is equally likely to go to any of the 12 users (note that it is possible that some users may not actually receive any email from us). What is the probability that the 20 emails are distributed such that there are 4 users who receive exactly 2 emails each from us and 3 users who receive exactly 4 emails each from us?

8. Say a hacker has a list of n distinct password candidates, only one of which will successfully log her into a secure system.
 - a. If she tries passwords from the list at random, deleting those passwords that do not work, what is the probability that her first successful login will be (exactly) on her k -th try?
 - b. Now say the hacker tries passwords from the list at random, but does **not** delete previously tried passwords from the list. She stops after her first successful login attempt. What is the probability that her first successful login will be (exactly) on her k -th try?
9. To get good performance when working binary search trees (BST), we must consider the probability of producing completely degenerate BSTs (where each node in the BST has at most one child).
 - a. If the integers 1 through n are inserted in arbitrary order into a BST (where each possible order is equally likely), what is the probability (as an expression in terms of n) that the resulting BST will have completely degenerate structure?
 - b. Using your expression from part (a), determine the smallest value of n for which the probability of forming a completely degenerate BST is less than 0.001 (i.e., 0.1%).

2 Coding

Download the starter code `cs109_pset1.py` and implement the function `play_game` as described in it. You **MUST** use the starter code. Submit your code to the “Gradescope PSet 1 - Code” portal. The Gradescope autograder will then report your problem score; there are no hidden tests. No partial credit will be given if you do not pass the autograder, but you can submit an unlimited number of times. We will only use the score from your most recent submission.

For code-writing questions, follow the below guidelines so that your code works with our autograder:

- Do not modify the name or signature of the `play_game` function, though you may use helper functions if you wish.
 - You will write code in the file `cs109_pset1.py`, which you can download from the course website. Submit only that file, and do not modify the name of that file. Do not submit a zip file.
 - Do not use global variables.
 - Follow all of the instructions that are provided as comments in the starter code.
 - We provide a file named `main.py` that you can run to execute the `play_game` function. We recommend running this file locally first and submitting to the autograder once your code runs locally.
10. Consider a game, which uses a generator that produces independent random integers between 1 and 100, inclusive. The game starts with a sum $S = 0$. The first player adds random numbers from the generator to S until $S > 100$, at which point they record their last random number x . The second player continues by adding random numbers from the generator to S until $S > 200$, at which point they record their last random number y . The player with the highest number wins; e.g., if $y > x$, the second player wins. Write a Python 3 program to simulate 100,000 games and return the estimated probability that the second player wins.