CS3000: Algorithms & Data — Summer 2023 — Laney Strange

Homework 4 - Short
Due Thursday June 22 @ 9pm Gradescope

Name:
Collaborators:

- Put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Thursday June 22 @ 9pm Gradescope. You may submit up to 48 hours late for no penalty, but expect a delay in grading.
- This homework cannot be used as one of your second-chance submissions.
- Solutions must be typeset, preferably in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class, is strictly forbidden.

Problem 1. *Shuffle (4 points)*

Write a new version of the Shuffle function from recitation that fixes both problems we saw in recitation: i.e., it generates all permutations with equal likelihood, and doesn't leave any permutations out.

Solution:

Problem 2. *RandSelect (4 points)*

Recall that we defined a balanced split for RANDSELECT as two partitions with at most $n/4$ and $3n/4$ elements, respectively. All other partitions would be unbalanced. The worst-case time for Rand Select with a balanced partition is $T(n) \leq T(3n/4) + cn$. The worst-case run-time for Rand Select with an unbalanced partition is $T(n) \leq T(n) + cn$. Compute the expected run-time for this algorithm and give its bound.

Solution:

Problem 3. *Complexity (4 points)*

A famous NP-Complete problem, which you'll very likely study in CS3800, is called *3-SAT*, aka "3-satisfiability". In 3-SAT, you have exacatly three truth values per clause, strung together into a complex statement.

This problem is about the easy one, though – 2-SAT. Find an assignment of truth values that would make the expression below TRUE or explain why one doesn't exist.

$$(a \vee \neg b) \wedge (\neg b \vee c) \wedge (\neg a \vee d) \wedge (\neg d \vee \neg c)$$

Solution: