Homework 1

Instructions

- The homework is due on **Friday 9/29 at 5pm ET**.

- No extension will be provided, unless for serious documented reasons.

- **Despite having two weeks, better start early than late!**

- Unless specified differently, the points are distributed equally among the sub-questions.

- Study the material taught in class, and feel free to do so in small groups, but the solutions should be a product of your own work.

- This is not a multiple choice homework; reasoning, and mathematical proofs are required before giving your final answer.

- If you work with others or utilize any external tools and resources, please make sure to annotate your answers.

- Please submit your work through Gradescope. You can find the access code on Piazza.

Exercise 1 [20 points]

A d-ary heap is like a binary heap, but (with one possible exception) non-leaf nodes have d children instead of 2 children. Let A be the array that represents the max heap.

a. How would you represent a d-ary heap in an array?

b. What is the height of a d-ary heap of n elements in terms of n and d?

c. Give an efficient implementation of EXTRACT-MAX in a d-ary max-heap. Analyze its running time in terms of d and n.

d. Give an efficient implementation of INSERT in a d-ary max-heap. Analyze its running time in terms of d and n.

e. Give an efficient implementation of INCREASE-KEY(A, i, k), which flags an error if $k < A[i]$, but otherwise sets $A[i] = k$ and then updates the d-ary maxheap structure appropriately. Analyze its running time in terms of d and n.

Exercise 2 [10 points]

Decide whether you think the following statements are true or false. If a statement is true, give a short explanation. If it is false, give a counterexample.

(a) True or false? In every instance of the Stable Matching Problem, there is a stable matching containing a pair (m, w) such that m is ranked first on the preference list of w and w is ranked first on the preference list of m.

(b) True or false? Consider an instance of the Stable Matching Problem in which there exists a man m and a woman w such that m is ranked first on the preference list of w and w is ranked first on the preference list of m. Then in every stable matching S for this instance, the pair (m, w) belongs to S.

Exercise 3 [25 points]

Gale and Shapley published their paper on the Stable Matching Problem in 1962; but a version of their algorithm had already been in use for ten years by the National Resident Matching Program, for the problem of assigning medical residents to hospitals. Basically, the situation was the following. There were m hospitals, each with a certain number of available positions for hiring residents. There were n medical students graduating in a given year, each interested in joining one of the hospitals. Each hospital had a ranking of the students in order of preference, and each student had a ranking of the hospitals in order of preference. We will assume that there were more students graduating than there were slots available in the m hospitals.

The interest, naturally, was in finding a way of assigning each student to at most one hospital, in such a way that all available positions in all hospitals were filled. (Since we are assuming a surplus of students, there would be some students who do not get assigned to any hospital.)

We say that an assignment of students to hospitals is stable if neither of the following situations arises.

- First type of instability: There are students s and s', and a hospital h, so that:
    - s is assigned to h, and
    - s' is assigned to no hospital, and
    - h prefers s' to s

- Second type of instability: There are students s and s', and hospitals h, h', so that:
    - s is assigned to h , and
    - s' is assigned to h'
    - h prefers s' to s
    - s' prefers h to h'

So we basically have the Stable Matching Problem, except that (i) hospitals generally want more than one resident, and (ii) there is a surplus of medical students.

Show that there is always a stable assignment of students to hospitals, and give an algorithm to find one.

Exercise 4 [10 points]

For this problem, we will explore the issue of truthfulness in the Stable Matching Problem and specifically in the Gale-Shapley algorithm. The basic question is: Can a man or a woman end up better off by lying about his or her preferences? More concretely, we suppose each participant has a true preference order. Now consider a woman w.

Suppose w prefers man m to m', but both m and m' are low on her list of preferences. Can it be the case that by switching the order of m and m' on her list of preferences (i.e., by falsely claiming that she prefers m' to m) and running the algorithm with this false preference list, w will end up with a man m'' that she truly prefers to both m and m'?

Resolve this question by doing one of the following two things:

- Give a proof that, for any set of preference lists, switching the order of a pair on the list cannot improve a woman's partner in the G-S algorithm; or

- Give an example of a set of preference lists for which there is a switch that would improve the partner of a woman who switched preferences.

Exercise 5 [20 points]

1. (6 points) Suppose that X is a random variable whose entropy $H(X)$ is 100 bits. Suppose that $Y = f(X)$ where f is a bijection between the range of X and Y.

   (a) What is the entropy of Y?
   (b) What is the joint entropy $H[X, Y]$ of random variables X, Y?
   (c) Consider $Z = g(X)$ where g is not a bijection. Assume there are at least two values $x_1, x_2 \in range(X)$ that map to the same $z \in range(Z)$, i.e., $g(x_1) = g(x_2) = z$. Does the entropy of Z equal 100, exceed it, or fall below it?

2. (6 points) Samuel Morse (1791-1872), the creator of Morse code, required this information to assign the most straightforward codes to the letters used most often. He determined this by merely tallying the letters in sets from printers' type. The data he derived are shown in Table 1, see also `https://www3.nd.edu/~busiforc/handouts/cryptography/letterfrequencies.html`. Consider X the random variable of a letter chosen according to the empirical frequency distribution of Table 1.

   (a) What is the entropy of the variable? You may (and should) code it up.
   (b) How does the entropy compare to the entropy of the uniform distribution on 26 letters?

Count	Letters
12 000	E
9 000	T
8 000	A, I, N, O, S
6 400	H
6 200	R
4 400	D
4 000	L
3 400	U
3 000	C, M
2 500	F
2 000	W, Y
1 700	G, P
1 600	B
1 200	V
800	K
500	Q
400	J, X
200	Z

Table 1: The frequency of the letters of the alphabet in English according to Morse's original dataset.

    (c) Why did Morse show interest in these specific statistics while developing his code?

3. (8 points) Prove the following facts on entropy:

    (a) (2 points) $H[X, Y] = H[X] + H[Y|X]$

    (b) (3 points) $H[X_1, \ldots, X_n] \leq \sum_{i=1}^{n} H[X_i]$. Can you give some intuition behind this inequality?

    (c) (3 points) Let X be a random variable that takes on values in the infinite set $\{1, 2, 3, \ldots\}$. Determine the entropy $H[X]$ when the probability distribution is given by $\mathbb{P}(X = k) = \frac{1}{2^k}$ for each k in $\{1, 2, \ldots\}$.

Exercise 6: Exhaustive Average Case Analysis [15 points]

The goal of this exercise is to perform *average case analysis* exhaustively. Specifically, given an instance of the Stable Marriage problem with 4 men and 4 women what is the average number of proposals men are going to make till they find their final wife in the Gale-Shapley algorithm? To do that, you will generate all possible instances of the Stable Marriage problem for a given set of womens' preferences.

    (a) (5 points) How many problem instances of stable marriage problem exist? What if we have n men in general instead of 4?

(b) (10 points) Read the Python file available at the Git repo `https://github.com/tsourolampis/bu-cs630-fall23`. Your goal is to fill the code that returns a dictionary and the average number of proposals. The keys should be the number of proposals made by the G-S algorithm, and the value the number of instances that yield this specific number of proposals. For example d[k]=m, would mean that m problem instances resulted in k proposals in total.