Homework 2 – Due May 1st, 2017 on Canvas

Homework Guidelines

Please make sure you read the collaboration policy, and write the following as the first line of your homework: "I have read and agree to the collaboration policy. ⟨`Your name`⟩." Your homework will not be graded if you do not write this.

Collaboration policy Collaboration on homework problems is permitted, but not encouraged. You are allowed to collaborate with *at most two students enrolled in the class.* You must mention the name of your collaborators clearly on the first page of your submission. Even if you collaborate, you are *expected to write and submit your own solution independent of others*, and your collaboration should be restricted to discussions only. Also, you should be able to explain your solution verbally to the course staff if required to do so. *Collaborating with any one not enrolled in the class, or taking help from any online resources for the homework problems is strictly forbidden.*

The Computer Science Department of UCSC has a zero tolerance policy for any incident of academic dishonesty. If cheating occurs, consequences within the context of the course may range from getting zero on a particular assignment, to failing the course. In addition, every case of academic dishonesty will be referred to the student's college Provost, who sets in motion an official disciplinary process. Cheating in any part of the course may lead to failing the course and suspension or dismissal from the university.

How to submit your solutions Each problem must be **typed** up separately (in at least an 11-point font) and submitted in the appropriate assignment box on the Canvas website as a PDF file.

This means that you will submit 5 separate files on Canvas, one for each problem!

You are strongly encouraged, but not required, to format your problem solutions in LaTeX. Template HW files and other LaTeXresources are posted on the course webpage. LaTeXis a free, open-source scientific document preparation system. Most technical publications in CS are prepared using this tool.

You might want to acquire a LaTeX manual or find a good online source for LaTeX documentation. The top of each problem should include the following:

- your name,
- the acknowledgement to the collaboration policy,
- the names of all people you worked with on the problem (see the handout "Collaboration and Honesty Policy"), indicating for each person whether you gave help, received help or worked something out together, or "Collaborators: none" if you solved the problem completely alone.

Solution guidelines For problems that require you to provide an algorithm, you must give the following:

1. a precise description of the algorithm in English and, if helpful, pseudocode,
2. a proof of correctness,
3. an analysis of running time and space.

You may use algorithms from class as subroutines. You may also use any facts that we proved in class.

You should be as clear and concise as possible in your write-up of solutions. Understandability of your answer is as desirable as correctness, because communication of technical material is an important skill. A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error and because it is easier to read and understand. Points might be subtracted for illegible handwriting and for solutions that are too long. Incorrect solutions will get from 0 to 30% of the grade, depending on how far they are from a working solution. Correct solutions with possibly minor flaws will get 70 to 100%, depending on the flaws and clarity of the write up.

Assigned Problems

Exercises (Do not hand in) Chapter 4: 3,4 and 6, Chapter 5:1-3.

Following are the problems to be handed in, 25 points each. Maximum score for this homework is 100 points. We will take your best four attempted problems.

1. (**Divide and Conquer**, 2-page limit – your solutions should fit on two sides of 1 page).

   A Walsh-Hadamard matrix H_n is an $2^n \times 2^n$ matrix with each entry being -1 or $+1$ and $n \in \mathbb{Z}^+$, such that the (i,j)-th entry of $H_n[i,j] = \frac{1}{\sqrt{2^n}}(-1)^{i \circ j}$. Assume that the rows and columns of H_n are counted from zero, i.e., the left-topmost entry of H_n is $H_n[0,0]$. Here $i \circ j$ is the bitwise dot-product of the binary representation of i and j represented with n bits. For example, if $i = 7$ and $j = 5$, then $\mathbf{i \circ j = (1,1,1) \cdot (1,0,1) = 2}$. Following are couple of examples of Walsh-Hadamard matrices H_2 and H_3.

$$H_2 = \frac{1}{\sqrt{2^2}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \qquad H_3 = \frac{1}{\sqrt{2^3}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

   - Starting from the fact that $H_n[i,j] = \frac{1}{\sqrt{2^n}}(-1)^{i \circ j}$, show that

$$H_n = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix}$$

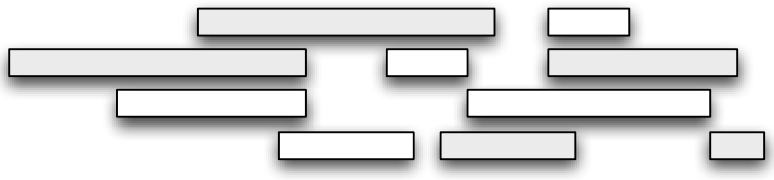
   - Show that the Euclidean norm of every column and every row is 1. (You are allowed to search Wikipedia for the definition of Euclidean norm.)

- Using the above properties, write and prove an induction claim that shows that the columns of H_n form an orthonormal basis, i.e., the dot-product of any two columns of H_n equals to zero, and every column of H_n has Euclidean norm of one. *Hint: You will need one of the properties of orthonormal matrices to show this.*
- Consider a vector $v \in \mathbb{R}^{2^n}$, i.e., a vector with 2^n entries with real numbers. Design an algorithm to compute $H_n \cdot v$ in time $O(n \log n)$. Prove the runtime bound for your algorithm.

2. (**Divide and Conquer**, 2-page limit – your solutions should fit on two sides of 1 page).

   You and your sister are travelling on a bus when you recall the "Hot and cold" game you used to play as kids. To kill time, you decide to play a version of it with numbers. One of you thinks of a number between 1 and n, and the other tries to guess the number. If you are the one guessing, every time you make a guess your sister tells you if you are "warmer", which is closer to the number in her head, or "colder", which is further away from the number in her head. Using this information you have to come up with an algorithm which helps you guess the number quickly. You can use a command called GUESS(x), where x is your guess, which returns "warmer", "colder" or "you guessed it!". You are required to give an English explanation for your algorithm. Also, prove the correctness of your algorithm and give an analysis of the space and time complexity. An ideal solution will propose an algorithm that takes $\log_2(n) + O(1)$ guesses in the worst case scenario. Recalling how binary search works might be helpful.

3. (**Greedy Algorithm**, 2-page limit – your solutions should fit on two sides of 1 page). The Menlo Park Surgical Hospital admitted a patient, Mr. Banks, last night who was in a car accident and is still in critical condition and needs monitoring at all time. At any given time only one nurse needs to be on call for the patient though. For this we have availability slots of all the nurses, which is a time from which they are available, a_i, to the time they have other engagements, b_i. You need to devise an algorithm that makes sure you can cover Mr. Banks' entire stay while having minimum number of nurses be on call for him. A nurse leaving at the same time as another arrives is acceptable. Following is an example of how the availability slots for the nurses will look like. The darker bars correspond to a set of 5 nurses who can cover the entire duration. (Notice, though, that 4 nurses would have sufficed.)

   Your efficient **greedy** algorithm should take input a list of pairs of times (a_i, b_i) for $i = 1$ to n

   (a) Consider the greedy algorithm that selects nurses by repeatedly choosing the nurse who will be there for the longest time among the periods not covered by previously selected nurses. Give an example showing that this algorithm does **not** always find the smallest set of nurses.

   (b) Present an algorithms that outputs a smallest subset of nurses that can cover the entire duration of Mr. Banks' stay at the hospital or say that no such subset exists.

   (c) Prove that your algorithm is correct.

   (d) State its running time.

4. (**Greedy Algorithm**, 2-page limit – your solutions should fit on two sides of 1 page). Picture, if you will, a long river with some towns scattered sparely along it. You are in charge of building a series of small hydro-electric power plants for these towns to give them some renewable electricity; however, the power plants can't power a town further than 20 miles away, due to their particular design - they can, however, give power to as many towns as they can reach. You want to build them along the river such that every town is within 20 miles of at least one of the plants.

   Give an efficient algorithm that achieves this goal using the minimum number of power plants. Prove using *greedy stays ahead strategy.*

5. [*] (**Optional, no collaboration**, 2-page limit – your solutions should fit on two sides of 1 page).

   You and your roommate are in a war over the thermostat - they like it cold, and you like it warmer. The temperatures range from 1 to n degrees. Your roommate is willing to leave the thermostat at some (unknown to you) temperature t or any lower temperature. You sit down with your roommate and agree to negotiate in the following way: In each round, you can name any temperture s between 1 and n. If $s > t$, they will say "too warm". Otherwise, they'll agree to s. Your goal is to ensure that the apartment is at the maximum acceptable temperture t.

   One way to ensure that you will have the temperture at s is to name all integers, starting from n and going down by 1 in each round, until they agree to the temperture t. But, if you follow this strategy, you might have to go through n rounds (irratating everyone involved).

   If you are allowed to change your mind (that is, ask for a higher temperature) after your roommate accepted your offer, you might want to try binary search as your strategy. However, it would also be annoying to change your mind so many times.

   (a) Suppose you are allowed to change your mind exactly once. Describe a strategy for ensuring a fair temperture, t, that uses $o(n)$ rounds of negotiation (as few as you can).

   (b) Now suppose you are allowed to change your mind k times, where $k > 1$. Describe a strategy for ensuring a fair temperture, s, with as few negations as you can. Let $f_k(n)$ denote the number of rounds you use, as a function of n. (The answer from part (a) is your $f_1(n)$.) For each k, you should be able to get an asymptotically better solution than for $k-1$: that is, make sure that $f_k(n) = o(f_{k-1}(n))$.