1. Jared the Meme Lord is back to separating his similar memes. Recall that Jared has a set of memes $M$, and he considers two memes $m$ and $m'$ to be similar if and only if there is a caption $c$ that is appropriate for both $m$ and $m'$. Also, the similarity relationship for memes is not transitive.

Jared wants to separate his memes into two groups so that he maximizes the number of pairs of similar memes that are separated.

Jared was satisfied with the results of Sadat’s randomized algorithm from before, but then he decided that he didn’t want to leave the state of his precious meme collection totally up to random chance. So Jared asked David, the authority on memes, if he could come up with an algorithm that provides a better guarantee. David proposed the following algorithm:

1. Arbitrarily partition the memes into two sets $A$ and $B$.
2. If there exists a meme $m \in M$ such that moving it to the other set in the partition increases the number of separated pairs of similar memes, do so.
3. Repeat step 2 until there is no such meme.

(a) Prove that the algorithm terminates.

(b) Prove that when the algorithm terminates, the number of separated pairs of similar memes is at least half the optimum.
2. Many Indians that I know like transporting food items – they carry food items wherever they go and bring food items back from wherever they travel to. During my recent visit to India, I brought back with me six packets of ground spices, one of which was unique. All the packets were labeled, but in a random security check, the TSA decided to check the contents of my bag and in the process all labels came off. I could not tell which one of the spice packets contained the unique spice as they all looked the same. On inspecting the packets more closely, I determined the probability \( p_i \) of the \( i \)th packet containing the unique spice to be given by \((p_1, p_2, p_3, p_4, p_5, p_6) = \left(\frac{8}{30}, \frac{7}{30}, \frac{5}{30}, \frac{5}{30}, \frac{3}{30}, \frac{2}{30}\right)\). The unique spice has a distinctive taste, so I decided to use the spices in cooking, one at a time, until I determined which packet contained the unique spice.

(a) To minimize the expected number of trials required to determine the unique spice packet, in what order should the spices be tried?

(b) What is the expected number of trials required?

(c) Since the distinctive taste of the unique spice is retained even when the unique spice is mixed with other spices (and being an adventurous cook), I mixed some of the spices and tried the mixture in my cooking. Encouraged by the outcome, I proceeded – mixing the spices, and testing the mixture in cooking, and stopped when the packet containing the unique spice had been determined. In expectation, what is the minimum number of trials required to determine the unique spice?

(d) Which mixture(s) of spices should be tested first?

(e) Is the strategy studied in (a) optimal, if we are allowed to mix the spices?

3. David, ecstatic after receiving his property from Sadat, decides to explore the house before he officially moves in. What he finds is mostly normal (i.e. khakras, printed salt bae memes), however he is surprised when he stumbles upon a closet filled with Sadat’s other hit game ”Battleshaik”. Curious to play the game, he opens each board game to instead see a pile of cash in each board game, sorted from the lowest dollar amount to the highest dollar amount. Strangely, this currency is not in U.S. dollars but some other strange currency which has unbounded positive values that can even be fractional!

David wants to take all this money home, however, he first wants to combine all of the stacks of money into one sorted stack. Given that there are \( k \) stacks of money, and \( n \) total bills, can you help David combine all of the money into one stack in \( O(n \log k) \) time?

4. (a) Using Huffman encoding scheme on a set \( S \) of \( n \) symbols with frequencies \( f_1, f_2, \ldots, f_n \), what is the longest a codeword could possibly be? Give an example set of frequencies that would produce this case.

(b) Prove that if some character occurs with a frequency more than \( 2/5 \), then there is guaranteed to be a codeword of length 1.

(c) If all characters occur with frequency less than \( 1/3 \), then there is guaranteed to be no codeword of length 1.
5. Deepan has invited all the 121 TAs to his brand new restaurant, ‘Heapify’! When the TAs sit down, they notice that the menu lists meals as a maximum heap according to price. The TAs feel like splurging, so they won’t buy any meal that costs less than $x$. Give an $O(k)$ algorithm to report all meals that cost more than $x$, where $k$ is the number of such meals, to help the TAs decide.

6. Prof. Stewart is consulting for the president of a corporation that is planning a technical event. Prof. Stewart is given a list of $n$ researchers to choose from and a list of all pairs of people who have collaborated on research with each other. We assume that the “collaboration” relation is symmetric, i.e., if person $A$ has collaborated with person $B$ then person $B$ also has collaborated with person $A$. Prof. Stewart has to invite as many people as possible, subject to the constraint: at the event, each person should have at least six other people whom they have collaborated with. Give an efficient algorithm that takes as input a list of $n$ people and a list of pairs of people who have collaborated with each other and outputs the list of invitees. Justify the running time of your algorithm?

7. As a real-estate mogul, Sadat has $n$ townships that he manages, and part of his management is to lay roads that connect various townships. Sadat must make sure that any township $x$ is reachable from any other township $y$. However, given that laying roads between townships is expensive, Sadat, as a shrewd business man and graduate of the Horton School of Funny Business, decides to connect two townships $x$ and $y$ with a road only if there exists no other set of roads that residents can traverse to get from township $x$ to township $y$ and vice versa (assume all roads are two-way). Given this connection scheme, Sadat wishes to find the two townships that are the farthest away from each other, where the distance between two townships $x$ and $y$ is the number of roads one needs to traverse to get from $x$ to $y$. Give an algorithm that runs on $O(n)$ time and returns a pair of townships that are the farthest away (i.e. the pair of townships with the largest distance between them).