Homework 7: Due March 26 (11:59 p.m.)

Instructions

- **Answer each question on a separate page.**
- **Honors questions are optional. They will not count towards your grade in the course. However you are encouraged to submit your solutions to these problems to receive feedback on your attempts. Our estimation of the difficulty level of these problems is expressed through an indicative number of stars** $('*' =$ **easiest**$)$ **to** $('*****' =$ **hardest**$)$**.**
- **You must enter the names of your collaborators or other sources as a response to Question** 0**. Do NOT leave this blank; if you worked on the homework entirely on your own, please write "None" here. Even though collaborations in groups of up to** 3 **people are encouraged, you are required to write your own solution.**

Question 0: List all your collaborators and sources: ($-\infty$ points if left blank)

Question 1: Binge Watching Greedily (5+15=20 points)

Garfield wants to watch TV **non-stop** for the entire time period $[S, F)$. He has a list of n TV shows (each on a different channel), where the i-th show runs for the time period $[s_i, f_i)$, and the union of all $[s_i, f_i)$ fully covers the entire time period $[S, F)$

Garfield doesn't mind switching in the middle of a show he is watching, but is very lazy to switch TV channels, so he wants to find the smallest set of TV shows that he can watch, and still stay occupied for the entire period $[S, F)$. Your goal is to design an efficient $O(n \log n)$-time greedy algorithm to help Garfield.

1. Describe your greedy algorithm in plain English. It is enough to provide a short description of the key idea for this part.
2. Describe how to implement your algorithm in $O(n \log n)$ time. Prove the correctness of your algorithm and the bound on its run time. (Hint: show that the output of your algorithm is never worse than *any* optimal solution.[1])

Question 2: Coin Change (2+8+6=16 points)

Given a set of coin denominations, consider the problem of making change for n cents using the fewest number of coins. Assume that the set of coin denominations consists of only positive integers.

1. Describe in plain English a greedy algorithm to make change consisting of the following coin denominations:

   $$\text{quarters } (25 \text{ cents}),\ \text{dimes } (10 \text{ cents}),\ \text{nickels } (5 \text{ cents}),\ \text{pennies } (1 \text{ cents})\ .$$

   For example if you are told to give out 118 cents, you can use 4 quarters, 1 dime, 1 nickel, and 3 pennies. It is not possible to use less than 9 coins in total.

[1] Given a list of n TV shows, there could be more than one optimal solution. You only need to return one of them.

2. Prove that your algorithm always outputs an optimal solution for the set of coin denominations given in the previous part.

3. Now we will show that the greedy approach is not necessarily optimal if a different set of coin denominations is used. Give a set of coin denominations for which the greedy algorithm does *not* output an optimal solution. Your set should include a penny (1 cent) so that there is a solution for any value of $n \in \mathbb{N}$.

Question 3: Huffman Codes (5+5=10 points)

1. In general, are Huffman codes unique? That is, given a set of letters and their corresponding frequencies, does any Huffman code-generating *algorithm* output the same encoding scheme? If yes, justify. Otherwise provide a counterexample. (Hint: different letters may have the same frequency in the general case.)

2. Is it possible that in an *optimal* code, a letter with lower frequency has a shorter encoding than a letter with a higher frequency? If no, explain why not. Otherwise, provide an explicit example of such code.

Question 4: Graphs (CLRS Appendix B.4, 2+2=4 points)

This problem is meant to refresh your memory on undirected graphs, which we shall use extensively in the second half of this course.

An undirected graph G is defined as a set of vertices V and a set of edges E which is a set of unordered pairs of vertices. For example, if $V = \{1, 2, 3, 4\}$ and $E = \{(1, 2), (2, 3), (3, 4), (1, 4), (1, 3)\}$ we have the graph shown below.

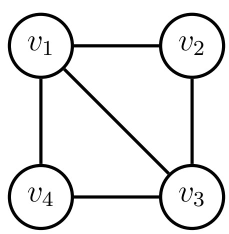


Figure 1: Illustration for Question 4

Are the following statements true for any undirected graph $G = (V, E)$? If true, provide a proof; if false give a counterexample. Here, $|V|$ denotes the number of vertices in the graph, and $|E|$ denotes the number of edges.

1. $\log |E| = O(\log |V|)$.

2. $\log |E| = \Omega(\log |V|)$.

Question 5: Trees (CLRS Appendix B.5, 4+6=10 points)

Let $G = (V, E)$ be an undirected graph. We say there is a *path* from $u \in V$ to $v \in V$ if there is a sequence of vertices $v_1, v_2, \ldots, v_k$ such that $(v_t, v_{t+1}) \in E$ for $t = 1, 2, \ldots, k - 1$ and $v_1 = u, v_k = v$. We say that

G is *connected* if for any two vertices $u, v \in V$ there is a path from u to v. A *cycle* is a list of vertices $v_1, \ldots, v_k$ where (v_t, v_{t+1}) is an edge for $t = 1, \ldots, k-1$ with $v_1 = v_k$.

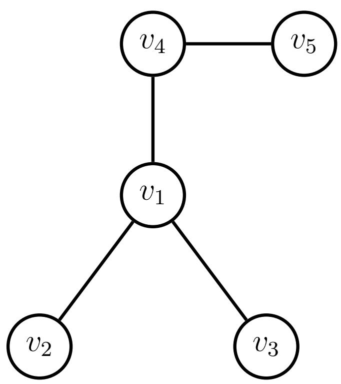
Recall that a *tree* is an undirected connected graph which does not contain any cycles (Figure 2a). Alternatively, an equivalent definition says that an undirected graph on n vertices is a tree if and only if it is connected and has exactly $n-1$ edges.

Let T be a tree on n vertices.

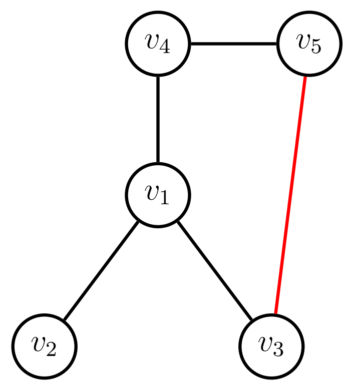
1. Suppose you add one more edge, call it e, to T (the number of vertices is kept the same). Call the resulting graph T'. Prove that T' contains a cycle. (See Figure 2b for an example).

2. Let C denote the cycle in T'. You now remove one edge, call it e', from C. Prove that the resulting graph, $T'' = T' \setminus \{e'\}$, is a tree. Note that T'' is not necessarily equal to T since *any* edge can be deleted from the cycle C (i.e., e' need not equal e).

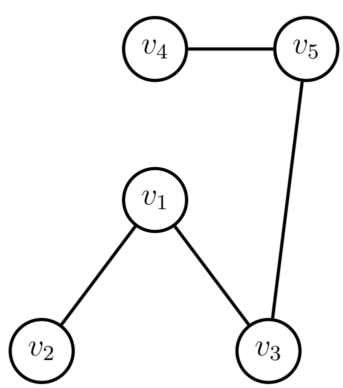
Hints for (2): We know that T'' has $n-1$ edges, so by the alternative definition of a tree, it is sufficient to show that T'' is still connected. For that, you'll need to show that for any two vertices u and v, there is a path in T'' from u to v. We know that there is a path between them in T' (why?); is this path also a path in T''? If not, how can we change it to become a path in T''?



(a) Example: Tree T on 5 vertices



(b) Example: T' is the tree T on 5 vertices with an additional edge (v_3, v_5) added. Notice that a cycle $(v_3, v_1, v_4, v_5, v_3)$ was formed.



(c) Example: We removed the edge (v_1, v_4) from T'. The resulting graph has no cycles but is still connected (thus it is a tree).

Figure 2: Illustrations for Question 5