Homework 2: Graph Search

Be sure to read the grading guidelines and style guidelines. Especially to see the suggested format for describing algorithms.

We sometimes describe how long are justifications or proofs are. These lengths are intended to help you estimate how much detail we're expecting; your proofs are allowed to be longer.

You are allowed (and encouraged!!) to collaborate with each other. Brainstorming is much easier to do in a group than alone! But you must follow the collaboration policy (which includes needing to write your submission on your own).

You will submit to gradescope; we will have a different box for each problem, so please give yourself extra time to submit.

1. Function Ordering [10 points]

Put these functions in increasing order. That is, if f comes before g in the list, it must be the case that $f(n)$ is $\mathcal{O}(g(n))$. Additionally, if there are any pairs such that $f(n)$ is $\Theta(g(n))$, mark those pairs. All logs are base 2.

- $f_1(n) = n^{n \log n}$
- $f_2(n) = (2n)^{n^2+1} - 999(\log(n))$
- $f_3(n) = (n + 99999)^{\log(99n)}$
- $f_4(n) = n^2$
- $f_5(n) = \log(n!)^2 + 3n + 20$
- $f_6(n) = 999 * \log(\log(\log(n)))$
- $f_7(n) = \frac{1}{100} \cdot \log(\log(\log(n))^{999})$
- $f_8(n) = n^{0.01}$
- $f_9(n) = \log(n)$

2. It's in the ears [25 points]

You have many pictures of alpacas and llamas, but none have been labeled with which species they are. Your friend, an alpaca expert, has repeatedly told you that if you look at the ears you can tell the difference. But you absolutely cannot remember what she said well enough to look at just one photo and tell whether it's an alpaca or llama.

What you can do, is given two photos say "those are the same species" or "those are different species." You do a bunch of such comparisons and write them all down. You then pick one of the photos and say "this animal is species A." Your goal is to perfectly place every animal into the buckets "species A" and "species B."

Given your data, you need to respond with one of three options:

- Inconsistent: you have analyzed the pictures in such a way that you can't possibly classify them all into species A and B correctly (for example you said photo 1 and 2 were the same species, photo 2 and 3 were different species, and photos 1 and 3 were the same species).
- Underspecified: your answers aren't inconsistent, but there is at least one picture that (from the data you've collected so far) could validly be either species A or species B.
- Exact answer: your answers aren't inconsistent, and every photo can be only one of species A or B.

Describe an algorithm to return which of those three cases the input matches, and if you're in the exact answer case to store the species labels for each photo.

Sample Input I: There are 4 images.
Image 1 and 2 are different species
Image 3 and 4 are the same species
Image 1 is "species A"

Correct Output: Underspecified

Sample Input II: There are 4 images.
Image 1 and 2 are different species
Image 1 and 4 are the same species
Image 3 and 4 are different species
Image 1 and 3 are different species
Image 1 is "species A"

Correct Output: Exact Answer

Sample Input III: There are 4 images.
Image 1 and 2 are different species
Image 1 and 4 are the same species
Image 3 and 4 are different species
Image 1 and 3 are the same species
Image 1 is "species A"

Correct Output: Inconsistent

(a) Give pseudocode (and/or English) to solve this problem. You will probably want to construct at least one graph, you can assume creating vertices and edges takes $O(1)$ time each, and you can give an English/mathematical description of the vertices and edges (don't worry about code constructs for creating it).

(b) Prove your algorithm is correct. You may use as a fact that any algorithm from this list works (as well as any analysis of any algorithm from class).

(c) Give the running time of your algorithm. For analyzing the running time, assume that you have p photos, s pairs identified as "the same" and d pairs identified as "different." Give a big-$\mathcal{O}$ bound in terms of p, s, d.

3. Not Regular Famous – Tik-Tok Famous [25 points]

You've started a TikTok account for your cat. Sadly, the algorithms haven't started spreading its cuteness...yet.

You decide the best solution is to send a DM to an influencer with your best video, knowing if it's cute enough they will share it with another person, who will share it with another, and another until your cat gets the virality it deserves.

You have a directed graph G. Every vertex is an influencer, and there is an edge from u to v if u can DM a video to v. For the video you have selected, you know that each influencer would (1) Tweet out the video to their followers (2) choose **exactly one** influencer in the graph to send a DM telling them to tweet the video too (u can send a DM to v if and only if there is an edge from u to v). That person would, in turn, both tweet the video to their followers and choose one person to DM, and so on.

Each vertex is labeled with a positive integer, corresponding to the number of views you will get **the first time** they tweet your video. Thereafter, if they receive your video via DM again, they will DM it again (possibly to a different person, possibly to the same one they did last time), but you don't (directly) get any new views.

Your goal is to determine the maximum possible views you could hope for.

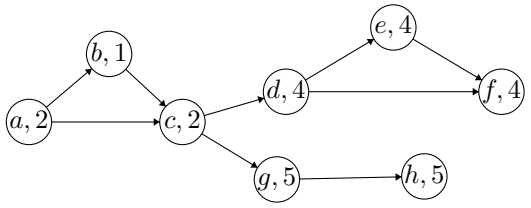
You are given a **directed** graph G, and a start vertex u, the first influencer you send your video to.

For this question, you do not have to prove your algorithms correct for any part. Instead, you should explain why you think it works in 2-3 sentences per part.

(a) Suppose G is strongly connected.

- Describe a walk (i.e., how the video will be DM'd between creators) that collects the maximum possible number of views (you don't need the shortest walk, just a walk).
- Describe an algorithm to find the number of views. (You don't need the algorithm to find the walk, just the number).

(b) Suppose G is a DAG. Describe an algorithm to find the maximum number of views. We recommend you consider the example below, where the best walk is: a, b, c, d, e, f. (Hint: you may find this part conceptually easier if you do a topological sort first! This step can be done in $O(V + E)$ time.)



(c) Now, stitch together the last two parts and describe an algorithm to handle any directed graph.

4. Tea Party [25 points]

You and a friend have a set of t teas you wish to try. Your tea-tasting guide warns you that, in order to get the optimal tea tastes, some teas are not safe to drink immediately after another (as you need time to cleanse your palate). Thankfully, the tea-tasting guide has provided you with a list of p pairs of teas, (t_1, t_2) where t_2 **can** be safely drunk immediately after t_1 without bittering the flavor (Note this **doesn't** necessarily mean you can drink t_1 after t_2. Order matters!)

You know that in a single tea-tasting session, you will drink exactly k teas, where a session will always start with English Breakfast and end with Earl Grey (as is proper tea drinking form). You wish to know how many different tea tasting sessions you could have. Specifically:

- The first tea must be English Breakfast, the last Earl Grey, and you must drink **exactly** k teas.
- You may repeat teas, but you must drink at least one other tea between repeated teas.
- Two sessions are different if there is at least one i, such that the i^{th} tea you drink is different (notice order does matter here! Changing the order of teas gives two different tastings).

Design an $\mathcal{O}\left((t+p)^k\right)$ algorithm to **count the number** of possible tea-tasting sessions. You only need to count the number, you don't need to have a full list of teas.

(a) Describe the algorithm. When building a graph, you can describe it in English and math rather than traditional pseudocode.

(b) Explain why your algorithm is correct. This does not need to be a formal proof, but should clearly describe why the things you find in the graph is the count of valid tea parties.

(c) State the big-$\mathcal{O}$ running time and justify it with 2-3 sentences. You may assume adding an edge to a graph takes constant time.

Hint: There are multiple different approaches here; we are imagining one that recursively walks through the graph, but other (clever) options are out there.