CPSC 320 2022W1: Tutorial Quiz 1, Individual Quiz Solutions

September 2022

It's a Job Seeker's Market (Quiz Version A)

You're in charge of matching CS undergraduate students with companies for summer jobs. By astonishingly good luck, you have n employers for n student applicants. The job market for software developers is booming: as a result, every employer desperately wants someone to fill their summer job spot, but **not** every student desperately wants to get a job, because they can likely find a job outside the department program if need be (or they can take summer courses, backpack through Asia, etc.). As a result, every employer has a complete preference list of all n students, but some students are unwilling to work for some employers and therefore have incomplete preference lists. A student would rather work for an employer on her preference list than be unmatched, but would rather be unmatched than work for an employer **not** on her preference list.

For example, for $n = 3$, a student s_1 may have the preference list $[e_2, e_3]$, which indicates that she is unwilling to work employer e_1. She would prefer to work for e_2 or e_3 than be unmatched, but would rather be unmatched than work for e_1.

An employer would prefer to be matched with **any** student than to be unmatched.

1. We will need to expand our definition of "instability" for this problem. The definition we saw in class still applies to the CS internship problem: namely, when e_i is matched with a student and s_j is matched with an employer on their preference list, but e_i and s_j prefer each other to their current partner. We can also consider it to be an instability when a student is matched with an employer not on his preference list: in this case, the student has an incentive to break the imposed matching (by quitting) and be happier as a result.

   Describe two new types of instability that can occur between: (a) an unmatched student and a matched employer; and (b) a matched student and an unmatched employer.

   The solution to this will be included in the solutions to Assignment 1.

2. We will now modify Gale-Shapley to solve SMP with the change that not every student needs to list every employer in their preference list.

   Here is the Gale-Shapley algorithm, with employers making offers:

   procedure GALE-SHAPLEY(E, S)
       initialize all employers in E and students in S to unmatched
       while an unmatched employer with at least one student on its preference list remains **do**
           choose such an employer $e \in E$
           make offer to next student $s \in S$ on e's preference list
           if s is unmatched **then**
               Match e with s $\triangleright$ s accepts e's offer
           else if s prefers e to their current employer e' **then**
               Unmatch s and e' $\triangleright$ s rejects e'
               Match e with s $\triangleright$ s accepts e's offer

```
        end if
        cross s off e's preference list
    end while
    report the set of matched pairs as the final matching
end procedure
```

Make a small change to the algorithm above to ensure that the (not necessarily perfect) matching produced never matches a student with an employer not in his preference list.

The solution to this will be included in the solutions to Assignment 1.

Breaking the Chain (Quiz Version B)

You're an administrator of a computer network. You want all computers in the network to be able to communicate with each other, and you also want to avoid a situation where one computer going offline breaks the communication path between other computers in the network.

It's common to represent networks as graphs, with each node representing a computer and edges between nodes representing the connections between computers in the network. You're interested in how the diameter of a network graph relates to the level of vulnerability of that network. Assume that your network is represented by a graph $G = (V, E)$, containing n nodes and m edges.

Extreme True and/or False

The next question(s) represents a scenario related to the previously stated problem domain and a statement about that scenario. Each statement may be **always** true, **sometimes** true, or **never** true. For each one, fill the circle by the best of these choices:

- If the statement is **ALWAYS** true, i.e., true in *every* instance matching the scenario.
- If the statement is **SOMETIMES** true, i.e., true in *some* instance matching the scenario but also false in some such instance.
- If the statement is **NEVER** true, i.e., true in *none* of the instances matching the scenario.

Then, **justify** your answer as follows:

ALWAYS answer: give and very briefly explain a small instance that fits the scenario for which the statement is true; and prove that the statement is true for all instances that fit the scenario.

SOMETIMES answer: give and very briefly explain a small instance that fits the scenario for which the statement is true; and give and very briefly explain a small instance that fits the scenario for which the statement is false.

NEVER answer: give and very briefly explain a small instance that fits the scenario for which the statement is false; and prove that the statement is false for all instances that fit the scenario.

1. **Scenario:** An undirected graph G with an even number of nodes $n \geq 4$ and diameter $n/2$, with diametric nodes denoted by s and t. **Statement:** there is a node v not equal to s or t such that removing v from G destroys all paths from s to t.

   Choose one: ○ **ALWAYS** ○ **SOMETIMES** ○ **NEVER**

   True instance (always/sometimes) *or* **proof that statement is false in all instances** (never):

   The solution to this will be included in the solutions to Assignment 1.

   False instance (sometimes/never) *or* **proof that statement is true in all instances** (always):

   The solution to this will be included in the solutions to Assignment 1.

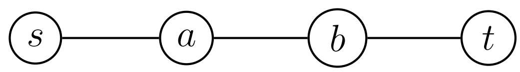
We have **partially completed** the Extreme True and/or False question below by filling in the appropriate Always/Sometimes/Never choice.

2. **Scenario:** An undirected graph G with an $n \geq 4$ nodes and diameter *strictly greater than* $n/2$, with diametric nodes denoted by s and t. **Statement:** there is a node v not equal to s or t such that removing v from G destroys all paths from s to t.

   Choose one: ● **ALWAYS** ○ **SOMETIMES** ○ **NEVER**

   True instance (always/sometimes) ~~*or* **proof that statement is false in all instances** (never)~~:

   Consider the graph with $n = 4$:

   The diameter of the graph is 3 (strictly greater than $n/2 = 2$), and eliminating either a or b from the graph removes all paths from s to t.

   ~~**False instance** (sometimes/never) *or*~~ **proof that statement is true in all instances** (always):

   NOTE: we consider this proof to be **too hard to reasonably ask for on a tutorial quiz**, so feel free to skip this portion of the question. However, you will need to prove this on your first assignment, so if you have some extra time you can work on it now!

   The solution to this will be included in the solutions to Assignment 1.

Extreme True and/or False (Quiz Version C)

Quiz contained instructions for Extreme True and/or False which are not repeated here.

1. **Scenario:** an SMP instance with $n \geq 2$ where all employers have identical preference lists. **Statement:** at least one applicant is matched with their first choice of employer in any stable matching.

   Choose one: ● **ALWAYS** ○ **SOMETIMES** ○ **NEVER**

   True instance (always/sometimes) *or* **proof that statement is false in all instances** (never):

   The example

   ```
   e1:  a1 a2      a1:  e1 e2
   e2:  a1 a2      a2:  e1 e2
   ```

   has a single stable matching $\{(e_1, a_1), (e_2, a_2)\}$, in which a_1 is with their top choice of employer.

   False instance (sometimes/never) *or* **proof that statement is true in all instances** (always):

   Proof that the statement is always true: If all employers have identical preference lists, this means all employers have the same first choice applicant, whom we'll call a_i. This applicant a_i must be with their first choice of employer in any stable matching. To show this, suppose that a_i's first choice of employer is e_j but they are instead paired with e_k. Then the pair (a_i, e_j) form an instability, because they aren't matched with each other but a_i prefers e_j to their current partner e_k (because e_j is a_i's top choice), and e_j prefers a_i over their current partner (because a_i is e_j's top choice). This matching is unstable, so a_i must be with their first choice of employer.

2. **Scenario:** an SMP instance with $n \geq 2$. **Statement:** there is exactly one stable matching.

   Choose one: ○ **ALWAYS** ● **SOMETIMES** ○ **NEVER**

   True instance (always/sometimes) *or* **proof that statement is false in all instances** (never):

   The example

```
e1:  a1 a2      a1:  e1 e2
e2:  a2 a1      a2:  e2 e1
```

has a single stable matching $\{(e_1, a_1), (e_2, a_2)\}$.

False instance (sometimes/never) *or* **proof that statement is true in all instances** (always):

The instance

```
e1:  a1 a2      a1:  e2 e1
e2:  a2 a1      a2:  e1 e2
```

has two stable solutions. The first (which we would obtain using Gale-Shapley with employers making offers) is $\{(e_1, a_1), (e_2, a_2)\}$. The second stable matching (which we would obtain if we ran Gale-Shapley with applicants making offers) is $\{(e_1, a_2), (e_2, a_1)\}$.