Davie Truong

“I have read and agree to the collaboration policy. Davie Truong”

Homework Heavy

Collaborators: Yona Edell

**Homework 1 Question 1 (Resident Matching)**

**Algorithm**:

Initialize each team and applicant to be free

While (positions are not filled) {

Choose any team with an open position

Applicant = current applicant that the cursor is on (assuming a linked list implementation for

T-Team’s Preference)

If (applicant is free)

Assign team and applicant to be paired

Else if (applicant prefers team’ to their current team)

Assign team’ and applicant to be paired and free his position on team

Else

Applicant rejects the team’ offer

**Description**: This algorithm is like the Gale Shapley algorithm, the only difference is that it runs until all the positions are full, rather than until the team has exhausted all its preference.

**Proof of Correctness**:

Claim: All teams get filled with applicants

Proof: The loop doesn’t stop until all positions are filled and since there are more applicants than positions, it will always fill and end the loop.

Claim: No unstable Pair

Proof: (by contradiction)

Suppose T3-Mark is an unstable pair: each prefer each other to their current pair T1-Mark/T3-Dobb.

Case 1: T3 never approached Mark.

T3 Prefers his current match Dobb to Mark.

T3-Mark is stable

Case 2: T3 approached Mark

Mark Rejects T3

Mark Prefers his team to T3

T3-Mark is stable

In Either case T3-Mark is stable.

**Time Complexity**:

Worst case the number of open position is one less total number of applying students (m=number of applying students) (as per conditions). Best case in that situation would be (m-1) assuming all positions fill with no need to revisit a position after a stability swap. However, if positions reopen after stability swaps, worse case the teams could have to go through every student. Thus (m-1) \* m meetings. Resulting in a time complexity of m^2.

**Space Complexity**:

Assuming this is done with link list and arrays, the space complexity would be (T X S) team by student array for the student’s preference for teams and an object with t number of linked list to keep track of preference and cursor for the list’s traversal of students as well as the team’s open position status.