CS216 Assignment1

12111012 匡亮

1. Problem

[Link to sustech space]

2. Algorithm Description

Initially all $s \in S$ and $c \in C$ are free

While there is a student s who is free and hasn't applied for all colleges

Choose such a student s

Let c be the highest-ranked college in s's list which he hasn't applied for

If $s_{s,c} > 0 \land c_{c,s} > 0$ then

If c is not full then

Add s to c's student list $student_c$

Else if $c_{c,s} > \min_{s' \in student_c}\{c_{c,s'}\}$ then

Remove s' from $student_c$

s' becomes free

Add s to $student_c$

Else then

s remains free

Endif

Endif

Endwhile

Return the set of list $student$

3. Time Complexity Analysis

Assume there are n students and m colleges.

In the worst case, all the students will try to apply for all colleges, and each try cost $O(\log n)$ time as we need to use a priority queue to maintain the worst student of each college. Therefore, the total time complexity is $O(mn \log n)$.

4. Stability Definition

The definition below referred to Lecture01 P30.

A matching S in this problem is **unstable** if there exists college c and student s such that all the following holds:

- $c_{c,s} > 0$ and $s_{s,c} > 0$;
- Either c is not full, or c prefers s to at least one of its enrolled students;
- Either s is unmatched, or s prefers c to the college he has applied for.

Also a matching S in this problem is **unstable** if for some $(s, c) \in S$ that $c_{c,s} < 0$ or $s_{s,c} < 0$.

5. Correctness Proof

Obviously we will never match a pair (s, c) that $c_{c,s} < 0$ or $s_{s,c} < 0$.

If there exists college c and student s makes S unstable, then there are two possibilities.

First, s has never applied for c. In this case, s is not free at the end, and has been accepted by some other college c', which means $s_{s,c'} > s_{s,c}$ as s applied for c' earlier than c.

Second, s used to be accepted by c but was refused later. In this case, as the student was once kicked out of the list, all the students in the list, including the students joined later, have a higher rank than him.

The two possibilities above are both contradict with (s, c) makes S unstable. Therefore, S is stable.

6. Optimality Anlysis

The anlysis below referred to Lecture01 P26-28.

Claim. The matching algorithm in this problem is **student-optimal**.

Proof. Let's prove by contradiction. Suppose y is **the first** student refused by a valid college, and a is **the first** college to do so.

When a rejected y, a has accepted a set of students $student_a$. Since they are never rejected by any valid college, they prefer a to any other valid college.

Let S be a stable matching in which a accepted y, call its student list now $student'_{a'}$, then for any $z \in student_a - student'_{a'}$, z prefers y to his college b, and y prefers z to a, therefore S is not stable.

Therefore, student will never be rejected by a valid college. As students apply for colleges in order of preference, students will always be accepted by their favorite valid college. So the matching algorithm gives us a **student-optimal** matching.