Extended Gale-Shapley Algorithm:

Lab01B variables restatement: N students, M colleges, i^{th} college with quota cap_i, i^{th} student's evaluation towards j^{th} college with $s_{i,j}$ (i=1,2,...,N; j=1,2,...,M), i^{th} college's evaluation towards j^{th} student with $c_{i,j}$ (i=1,2,...,M; j=1,2,...,N). (For convenience, we denote cap=max of cap_i)

1. What is the time complexity?

Answer: Intuitively, the time complexity of the input is O(MN). In preparation, we use merge sort to sort the $s_{i,j}$ in descending order respectively, which costs O(NMlogM). Then we add students unmatched into the waiting queue. In order, each student apply for the optimal college which hasn't been applied before,according to the ranking of students' evaluation towards colleges with $s_{i,j}>0$. For the college applied, if the number of the admitted students hasn't reached to its quota and the college's evaluation to this student is not less than 0, then it will accept this admission. If the college's evaluation towards this student is less than 0, then it will reject this student and we add this student into the waiting queue again. If the college's evaluation towards this student is greater than 0,but it has accepted enough applications, then it will compare this student to each student it has accept and if it prefers this student some student in its accepted list,then it will replace the student of

minimum evaluation in list with this student. All in all, what we need to do is to maintain a min heap of size cap_i for each college. So, the time complexity of building heap for each college is O(caplog(cap)). In the worst case, every student tries to apply for the i^{th} college and after cap_i^{th} application $c_{i,j}$ is always greater than the top-heap element, which means there will follow $(N-cap_i)log(cap_i)$ insertion by replace the top-heap element with new element and adjust its position in the heap. So, above all, the main part of the algorithm cost O(NMlog(cap)). Consequently, the time complexity of extended GS Algorithm to Lab01b is O(NMlog(max(M,cap))).

2. How do we define stability for the college/student setting?

Answer:

It's called stable matching if all of the followings are satisfied:

(1)No student prefers not to go to college than to reserve his/her current admission.

(2)No college prefers to abandon an enrolled student than to reserve him/her.

(3)There is not any pair (S,C), such that

a. student S is not enrolled to college C.

b. student S prefers college C than S's current admission state (i.e. being enrolled to some college, or unmatched)

c. either C is capable to enroll more student and S can increase C's satisfaction value, or C prefers S than some other student S’ that has been enrolled to C.

3.Prove that the extended GS algorithm indeed outputs a stable matching.

Answer: suppose there exists unstable matching after we run extended GS algorithm.

Case Ⅰ :some student prefers not to go to college than to reserve his/her current admission. It’s contradictory because every student will only submit his own application to the college he wants, which means $s_{i,j} > 0$.

Case Ⅱ :some college prefers to abandon an enrolled student than to reserve him/her. It’s contradictory because every college will only accept the application of the student whose $c_{i,j} > 0$.

Case Ⅲ :there exists a pair (S,C) such that S is not enrolled to college C and student S prefers college C than S's current admission state (i.e. being enrolled to some college, or unmatched) and either C is capable to enroll more student and S can increase C 's satisfaction value, or C prefers S than some other student S’ that has been enrolled to C. It’s contradictory because S submitted his application to all college in descending order of $s_{i,j}$,and provided that college is willing to and can contain this student, this student

will undoubtedly be enrolled to the first college which is willing to and can contain him when student traverses all colleges in descending order of $s_{i,j}$. So, he won't be encountered with another school which is willing to and can contain him, meanwhile, he prefers this school to his current admission.

4.Does the student-proposing GS algorithm produce a student-optimal matching?

Answer: Yes, it does. Suppose that student-proposing GS algorithm is not student-optimal, i.e, some student is not admitted by his favourite valid college. Since he submitted his application in descending order of $s_{i,j}$, so he must have been rejected by valid college before. Let S1 be the first student who was rejected by his favourite valid college before, and let C1 be first valid college in descending order of $s_{i,j}$ that rejected him. When S1 was rejected by C1, which means C1 accepted another student who also prefers C1, we denote this student as S2. In stable matching S, (S1,C1) $\in$ S, and we denote that (S2,CX) $\in$ S, so from above, C1 prefers S2 than S1 and S2 prefers C1 than CX, so (S1,C1) and (S2,CX) are both unstable. It's contradictory.