Qiyao Zhou

Z5379852

Question 3

3.1

First, we create two arrays X [1...n] and Y [1...n] to keep the serial number of class and labs.

Next, we can sort C and L in both O (n log n) in descending order, using Merge Sort. While sorting, we need to maintain X and Y so that their serial numbers are always at the same index as the corresponding numbers in C and L respectively.

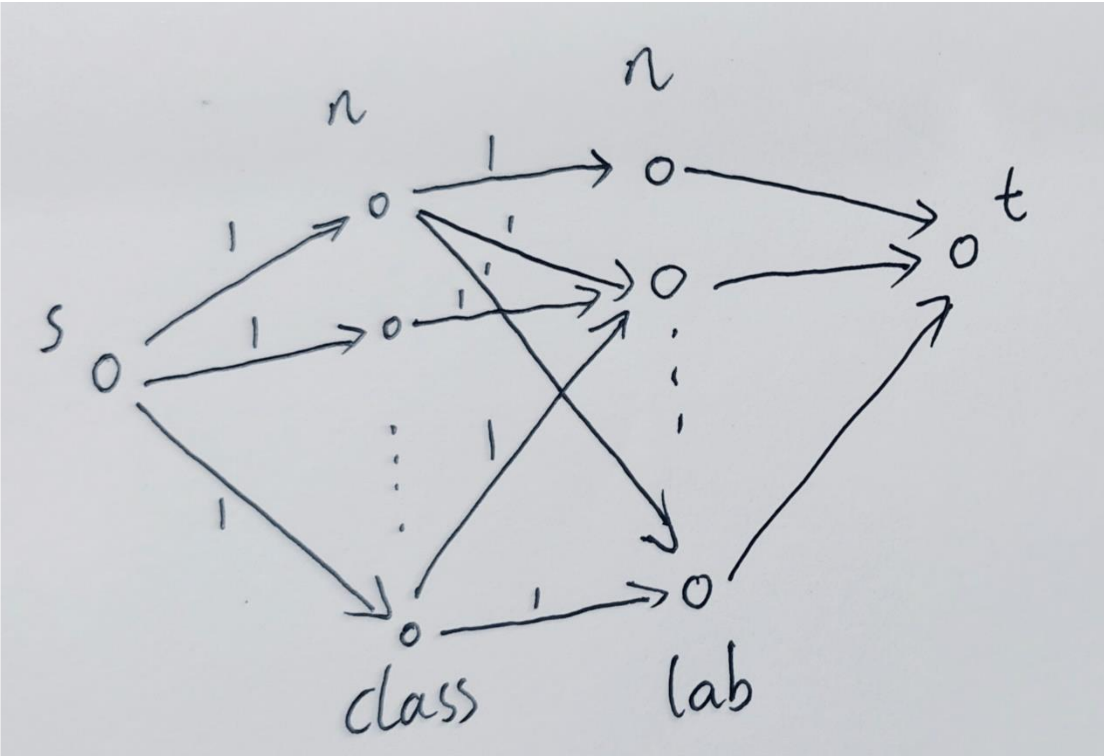
Then, according to the sorted C and L arrays, we can compare them one by one from the beginning, if $C_i > L_i$, the result is returned as impossible, otherwise the lab Y_i is assigned to class X_i.

In terms of time complexity, sorting requires 2O (n log n) and final matching requires O (n). The total time complexity is 2O (n log n) + O (n)= O (n log n).

3.2 We begin by constructing a flow network using the input data. The network is showed in the diagram below:

- source s and sink t,
- the combinations of vertices from left to right represent class and lab,

- connect s to each class vertex with capacity equal to 1,
- connect class vertices to lab vertices they want (B[i][j] =True) with capacity equal to 1,
- connect each lab vertex to t with capacity equal to 1,



From this flow network construction, we run Ford-Fulkerson to find the maximum flow. If the maximum flow is less than n, then we output "no solution". Otherwise, it is possible to assign a lab room to each class. We can deduce an assignment by examining which of the (C[i], L[j]) are in the max flow. If it is in the max flow, then we can assign lab room j to class i. The time complexity is O(|V||F|) where V= 2n+2 and F <=$n^2 + 2n$, so the algorithm runs in O (n^3).

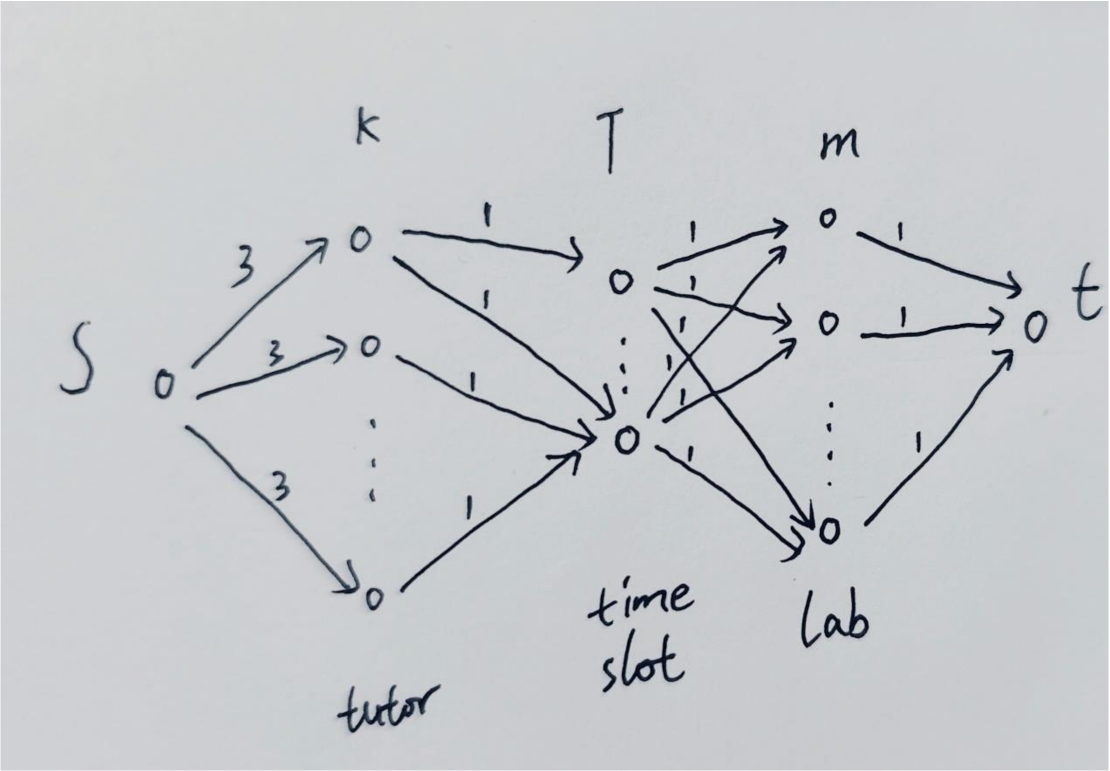
3.3

The solution to this problem is nearly the same as 3.2, so we can construct a network diagram almost identical to 3.2 and then solve it by the Ford-Fulkerson method. The only difference between the newly constructed network diagram and 3.2 is that the capacity between the lab vertex and t has been changed from 1 to T. If the maximum flow is less than n, then we output “no solution”. Otherwise, it is possible to assign a lab room to each class.

The time complexity is O(|V||F|) where V= m+n+2 and F <=mn+m+n, so the algorithm runs in O (n^3).

3.4 We begin by constructing a flow network using the input data. The network is showed in the diagram below:

- source s and sink t,
- the combinations of vertices from left to right represent tutor, time slot and lab,
- connect s to each tutor vertex with capacity equal to 3,
- connect tutor vertices to time slot vertices they want (D[i][j] =1) with capacity equal to 1,
- connect time slot vertex to each lab vertex with capacity equal to 1,
- connect each lab vertex to t with capacity equal to 1.



From this flow network construction, we run Ford-Fulkerson to find the maximum flow. If the maximum flow is less than m, then we output “no solution”. Otherwise, it is possible to assign a room, time, and tutor to each lab. We can deduce an assignment by examining which of the (tutor, timeslot, lab) are in the max flow. If it is in the max flow, then assign time and tutor to the lab.

The time complexity is O(|V||F|) where V= k+T+m+2<3n+2 and F $=kT + mT + k + m < 2n^2 + 2n$, so the algorithm runs in O (n^3).