The class scheduling problem

Introduction

The computer science and software engineering department at Concordia University has S students and offers C courses. It also has access to N classrooms and M weekly time slots. Every student is enrolled in multiple courses and we can assume that each course has exactly one weekly lecture, takes exactly one time-slot, can take place in any of the classrooms and has at least one student enrolled in it.

A valid scheduling is an assignment that maps to each course a classroom and a weekly time slot such that there are no conflicts (i.e. if two courses have students attending both of them they cannot be scheduled at the same time). For instance, even if the department has 5 classrooms available, a scheduling might only need to use 4 classrooms. A classroom is used if at least one lecture per week is held in that room.

The problem is to find a valid schedule.

Questions

1. Model this problem using graph theory.
2. Propose a Greedy algorithm that finds a valid scheduling if one exists.
3. Proof its correctness. (HINT: prove first that it is valid and after that it is correct.
4. Implement your algorithm as specified below.

Specifications

The input is specified in a file called *input.txt* as follows: the first line contains an integer T specifying how many data sets are included in the file. The remaining file contains all these data sets in sequence. Each data set starts with a line consisting of 4 space separated positive integers S, C, N and M. The following block of S lines describes the enrollment Each of these lines contains a sequence of space separated integer numbers: . The interpretation of this line is that the student takes courses denoted by where.The ordering of is arbitrary.

Here is an example:

3

3 3 3 2

0 1 0

1 1 1

2 1 2

3 3 3 2

0 3 0 1 2

1 3 0 1 2

2 3 0 1 2

3 3 3 2

0 2 0 1

1 2 0 1

2 2 1 2

In all these 3 examples there are 3 students, 3 courses, 3 classrooms and 2 time-slots. In the first one each student takes one course and any schedule would be valid as there are no possible conflicts. In the second example every student takes all courses offered and no valid schedule can be found as no two courses can be scheduled at the same time and we only have 2 time-slots available. In the third example, a schedule can be found, for instance course 1 is scheduled in classroom 0 at time-slot 0, course 1 is schedule in classroom 0 at time-slot 1 and course 2 is scheduled in classroom 1 at time-slot 1. Note that in this case the time-slot used by course 1 cannot be used by any other courses as it will result in a conflict.

The output is a file called *output.txt* and contains one line for every input. The first character is a Y if a solution exists, and N if no solution exists. If a solution exists, the second element is a number is the minimum number of classroom needed. Finally, the following numbers are a sequence of pairs each denoting the scheduling of the course i: denoting the classroom and denoting the time slot. All these number are separated by spaces.

Here is a possible output for the file above:

Y 2 0 0 0 1 1 0

N

Y 2 0 0 0 1 1 1

Answers:

1. Each node represents a course if at least one student registered. Each edge between two nodes represents there is at least one student registered these two course(node) at the same time
2. Sort the degree of node in decreasing order, fill the table

e.g.: for 3rd dataset

Decreasing order of nodes(courses): Co1, Co0, Co2

Fill process:

1. Fill co1 to s1
2. Co1 is connected to Co0, then fill Co0 in another slot, which is S2
3. Co2 is connected to Co1, the Co2 cannot be filled into S1(because Co1 is in S1), so fill it in S2.

|  |  |  |
| --- | --- | --- |
| Room 3 |  |  |
| Room 2 |  | Course 2 |
| Room 1 | Course 1 | Course 0 |
|  | Slot 1 | Slot 2 |

