COMP3010 - Algorithm Theory and Design

Assignment 1 - Creating a National COVID-19 Advisory Committee

```
Worth: 20%  overall unit
Due: 8pm Sunday 5 September 2021 on iLearn.
```

*Objectives: The purpose of this assignment is to test your ability to design **efficient** algorithms, analyse them, and implement them*. Your programs MUST work correctly in JAVA (Java 16). This assignment also tests your ability to understand standard specifications and follow instructions carefully. Failing to submit the assignment as required will cause you to fail the assignment.

Overview

The National Cabinet of the Prime Minister meet regularly to discuss Australia's COVID-19 response, recent outbreaks of COVID-19 and the Australian COVID-19 Vaccine Strategy. As the situation remains complex, it has been agreed a National COVID-19 Advisory Committee is set to advise the National Cabinet. The Prime Minister (PM) first needs to guarantee that the new National COVID-19 Advisory Committee represents all stakeholders (States/Territories governments, Medical Experts, Industry Vaccin Experts, Safety and Security Experts,...). With the large number of eligible groups, the PM struggles to select enough appropriate members while keeping the size of the committee under a number which would allow constructive recommendations.

You have been hired by the PM to write a program that will select *at least one member in each given group while minimising the total number of members* in the committee.

For this assignment, you are given a simpler version of this problem: (a) there are g groups (although g can be arbitrarily large, i.e. you cannot assume it is bounded) and (b) each member may appear in several groups but is identified by a unique positive number. However, you cannot make any assumptions on the properties of the members ids (i.e., it can be ANY positive integer) and the number and size of groups (i.e., you CANNOT assume a maximum number or a maximum size).
A solution is a set of members such that each group has at least one member in this solution set.

Your Work.

For the following problem, your program must ask the user for:

1. On the first line, the number of representative groups (i.e., an integer followed by end-of-line), then
2. on each following lines, each corresponding to a group, the list of ids of its members: i.e., a sequence of integers, separated by spaces, terminated by the number "0" followed by end-of-line.

Your task is to find a solution that minimises the number of members selected in the committee such that at least one member of each group is present in the committee. You must output:

1. On the first line: the number of members selected, then
2. on the second line, the list of ids of the members selected by your solution (in any order), **followed by an end-of-line**.

Be careful not to output anything else (e.g., debugging comments,...). You can assume that the user will not try to crash your program by entering clearly invalid data (characters or negative input, members with id = 0, empty groups, more groups than initially mentioned,...). You must at least submit a main program in a file called *ass1_comp3010.java*. However, you can create as many other functions and create other files if needed (make sure you submit them). I will NOT be running your program in Eclipse or any other IDE: I will be compiling your Java code, and running it using a script. That is, I will be compiling it (*javac ass1_comp3010.java*) and running it using *java ass1_comp3010*. The program should just print out its output in the usual way, using e.g. *System.out.println()*. **Please note that, as your program will be also tested automatically you must comply to the specifications**. In particular, pay attention to the required end-of-line, number "0" at the end of each group,...

An example executable solution, *ass1_sample* will be shown in the Lectures to help you understand the input/output requirements. However, please note that, as you can see clearly with the first example, this solution is not optimal (i.e., does not select the minimum number of members as it could be done by selecting member 5 only). In fact, it will often give you a solution far from the minimum. So do not believe in the quality of your program is optimal if it gives similar or close answers.

```
> java ass1_sample
Enter the number of groups from which you must find representatives: 5
Enter the list of members of each group (one group per line, each terminated by 0):
1 2 3 4 5 0
2 3 4 5 6 0
3 4 5 6 7 0
4 5 6 7 8 0
5 6 7 8 9 0

The number of members selected and their ids are :
5
1 2 3 4 5
> java ass1_sample
Enter the number of groups from which you must find representatives: 5
Enter the list of members of each group (one group per line, each terminated by 0):
1 2 3 4 5 6 7 8 9 0
2 4 6 8 0
1 3 5 7 9 0
2 3 5 7 0
4 6 8 9 0

The number of members selected and their ids are :
3
1 2 4
> java ass1_sample
Enter the number of groups from which you must find representatives: 3
Enter the list of members of each group (one group per line, each terminated by 0):
1 2 3 0
2 3 1 0
3 1 2 0

The number of members selected and their ids are :
1
1
> java ass1_sample
Enter the number of groups from which you must find representatives:
10
Enter the list of members of each group (one group per line, each terminated by 0):
12 21 0
123 234 0
1234 5432 0
3456 12 0
12345 234 0
234 123456 0
123456 1234567 0
12345678 12345 0
908070605 123 0
21 123 0

The number of members selected and their ids are :
6
12 123 1234 12345 234 123456
```

Your task for both parts

1. Submit a file *ass1_comp3010.java* (and other separate files if necessary - no zip files) that use an EFFICIENT algorithm to provide the best output as possible. By efficient, we mean that your program must use a low complexity (both in time and in memory-size) AND that the number of members selected is as small as possible. ALL programs should be clearly presented (e.g., including invariants, pre- and post-conditions when needed) with appropriate style, presentation, clarity, lack of typos.
2. For each part, you must submit a report/portfolio/blog (word or pdf are both acceptable) summarising the different steps of your attempts (what works and what did not), and reflecting on the things you learnt along the way. It must include one page describing the efficiency analysis of your program based on the input (it is up to you to determine what the important input here) using asymptotic notations (at least big-Oh notation, and, if appropriate, big-Omega and big-Theta). All portfolios/reports do not need to be long, 2 pages maximum, but they must be clear on where you are at.

Your Work.

You must work on your own and must write your program **ass1_comp3010.java** (14%) and analysis portfolio (6%).

Warning and Hints

- You may discuss general approaches to a solution, but **cannot discuss** specific approaches and **cannot show** algorithms and code to others.
- If you re-use any part of code which has been presented in the Lectures or is in a text book, make sure you cite the reference when you do so.
- For each input, there may be several optimal solutions.
- For finding the best "minimum" number of members, you will be competing among yourselves: the quality of your solution will be compared to the best solution find by one of you.
- If your solution is too far from the minimum and your algorithm is inefficient, you will end-up with less than 50%.

Submission of Assignments

Your program (with other files if needed) and portfolio MUST be submitted electronically using iLearn as requested (NO paper or email will be accepted).

That is, for the programming part, you must submit to iLearn at least one Java file, that is: *ass1_comp3010.java* compiling correctly including the corresponding main program.

Frequently Asked Questions (FAQ)

I'll attempt to answer any questions about the assignment sent to me by email within 1 or 2 open days. Although if your question is genuinely of interest, an answer will be posted in the relevant iLearn discussion page by the next working day, so that everyone can benefit from it. So please, read the FAQ before asking. **You are expected to read the FAQ regularly (and at least before the submission deadline).** This assignment also tests your ability to understand standard specifications and follow instructions carefully. Failing to submit the assignment as required might cause you to fail the assignment.