COT 4400: Analysis of Algorithms
Final Project

You are not allowed to use the internet or consult any references. The only people you can work with on this project are your group members. This policy is strictly enforced.

This project requires you to develop a heuristic for an NP-Hard problem[1] and then write a program that implements your solution. Note that the problem is NP-Hard, so don't plan on getting a perfect solution. Your task is going to be to get as good a solution as possible using a a reasonable amount of computer time. Please note that my test examples my contain a large number of records, so you should design an algorithm with polynomial complexity[2].

1. *Problem Description* A *combinatorial auction* is a particular mechanism developed by economists for selling a collection of items to a collection of potential buyers. (The Federal Communications Commission has studied this type of auction for assigning stations on the radio spectrum to broadcasting companies.)

   Here's a simple type of combinatorial auction. There are n items for sale, labeled $I_1, \ldots, I_n$. Each item is indivisible and can only be sold to one person. Now, m different people can place *bids*: The i^{th} bid specifies a subset S_i of the items, and an *offering price*, x_i that the bidder is willing to pay for the items in the set S_i, as a single unit. (For the problem description, we will represent this bid as pair (S_i, x_i).)

   An auctioneer now looks at the set of all m bids; she chooses to *accept* some of these bids and to reject the others. Each person whose bid i is accepted gets to take all the items in the corresponding set S_i. Thus the rule is that no two accepted bids can specify the sets that contain a common item, since this would involve giving the same item to two different people.

   The auctioneer collects the sum of the offering prices of all accepted bids. (Note that this is a "one-shot" auction; there is no opportunity to place further bids.) The auctioneer's goal is to collect as much money as possible.

   Therefore, the problem of *Winner Determination for Combinatorial Auctions* asks: Given items $I_1, \ldots, I_n$ and a set of bids (S_1, x_1),...,(S_m, x_m). Find the collection of bids that the auctioneer can accept so as to maximize the amount of money collected.

2. *Input/Output*

   - The Input (`input.txt`)

     The input file contains multiple sets of instances. The input file begins with a single positive integer on a line by itself indicating the number of instances following, each of them as described below. This line is followed by a blank line, and there is a blank line between consecutive instances.

     The first line of each instance contains two positive integers, separated by a space, where the first positive integer, n, denotes the number of items and the second positive integer, m, denotes the number of bids. Each of the next m lines represents a bid.

     A bid line contains a single positive integer, k, which is the amount the bidder is willing to pay for the products followed by l integers which indicate the items included in this bid. The amount and product indicies are separated by spaces.

   - The Output

     The output file should contain three lines for each instance.

     (a) The first output line should indicate the run time of the algorithm in milliseconds.
     (b) The second output line should indicate the total number of bids accepted and the total amount collected by the auctioneer for these bids (two positive integers separated by one space)
     (c) The third output line should indicate the index each accepted bid, where bids are indexed 1 through m.

     There should be a blank line between consecutive instances.

     Note that if more than one subset of bids yields the the optimal solution then any one will do.

   An example input/output file will be posted on Canvas.

[1] The problem has been adapted from Algorithm Design, by Kleinberg and Tardos.
[2] If your algorithm/code is too slow to handle the inputs you will lose points for both your algorithm design grade AND your results grade.

3. *Example*
   Suppose an auctioneer decides to use this method to sell 5 items $(I_1, I_2, I_3, I_4, I_5)$. The bids placed for these five items are:

   S_1 \$5: I_3, I_4

   S_2 \$1: I_1, I_2

   S_3 \$5: I_5

   S_4 \$10: I_1, I_2, I_3, I_4, I_5

   The auctioneer should choose the bids placed by S_1, S_2, and S_3 since they are all unique items and result in \$5+\$1+\$5=\$11 whereas choosing S_4 would block S_1, S_2 and S_3, but only result in \$10.

   The input for this instance would be:

```
5 4
5 3 4
1 1 2
5 5
10 1 2 3 4
```

   The expected output for this instance would be

```
0
3 11
1 2 3
```

Project Deliverables

Schedule a 20-minute meeting with me to demonstrate your project in one of the time slots that I will make available for this purpose. All group members must be present at this meeting. I will email input files in the format described above to all group members approximately twenty minutes before the scheduled time for your demo. *You will have to run your program on these inputs and output the results and the run time of your program in the given format.* The meeting will take place in my office (ENB-339). Be prepared to answer questions about your algorithm, its implementation, and what each group member's contribution was. The only written deliverable is the *outcomes* form that has been posted separately. Answer the questions on this form and use it to report your results. You will also electronically submit your code and an output text file for the input file.

1. [80 points] Algorithm Design, Analysis, and Implementation
   Develop a reasonable algorithm that is able to get as good a solution as possible using a a reasonable amount of computer time. Be able to discuss your algorithm, including design decisions, implementation, and analysis of your algorithm.

2. [20 points] Results
   A portion of your grade will be based on how your solution compares to solutions generated by other groups.

3. Submit group effort percentages and describe who did what. The effort percentages must be agreed upon by the group. See syllabus to determine how we will use this to compute your individual grade.
   Note that we reserve the right to change your group effort percentages. Group effort percentages will be changed if we think they do not accurately represent the contribution of each student.

Good Luck!!!