Fall 2009: CS 430 - Project Part 1 Version 1.0

Assigned October 23. Due November 6th.

The goal of the project is to implement several algorithms for Maximum Coverage and variants. This first part consists of implementing the basic **Greedy** algorithm.

Maximum Coverage

Input: A collection of m sets $S_1, S_2, \ldots, S_m$, all subsets of a universe U of n elements, and a number k.
Output: A subset Q of $\{1, 2, \ldots, m\}$ with $|Q| \leq k$. Q gives a subcollection of at most k sets.
Measure: Maximize $|\cup_{i \in Q} S_i|$, that is, the number of elements covered.

The Project

Teams of three students will work together on implementing and comparing algorithms for this problem and variants. One algorithm is the greedy algorithm for set cover that we discussed in class. **Greedy** only provides an approximate solution, and your job is to implement a very fast variant (which you may have to invent).

Other "approximation" algorithms from the literature will be discussed later.

Input and Output Formats

The programs should read the input from a sequence of files called "instance01" to at most "instance99", and output solutions in the files "solution01" to "solution99". Instances will have at most 1000 sets and at most 20000 elements.

Each input file starts with a line containing m, n, k, and p, followed by p lines each containing two integers: the ID i of a set and the ID j of an element, meaning set S_i contains element j. The convention is $U = \{1, 2, \ldots, n\}$, and m and k are in the description of Maximum Coverage. An example, with five sets: $S_1 = \{1, 2, 3, 4\}$, $S_2 = \{1, 2, 5\}$, $S_3 = \{3, 4, 6\}$, $S_4 = \{5\}$, and $S_5 = \{6\}$, is below:

```
5 6 2 12
1 1
1 2
1 3
1 4
2 1
2 2
2 5
3 3
3 4
3 6
4 5
5 6
```

The output will have a single line with $\min(k, m)$ integers: the IDs of the selected sets. Break ties arbitrarily (so the output is not unique).

Deliverable

Submit your source code (C++ or Java only) to blackboard. Also turn in a hard-copy of your pseudocode with running time analysys. Full marks for $O((m + k) \lg m + n + p)$ running time. Extra credit for $O(m + k + n + p)$.

Outside Sources

There should be no help from persons other than the instructor or the TA. Any collaboration in between teams is strictly prohibited,

You can use any open source for both ideas and code, provided you clearly mark when the idea or code is borrowed, and give full acknowledgement to the source. But make sure you are not breaking any laws.

Team composition

Please send me email by November 4 with the list of the members of your team. If you do not have a team, or your team does not have enough members, I will assign students to teams.