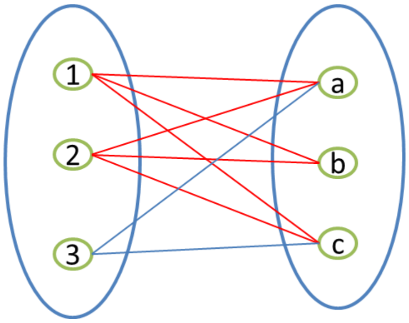
Homework #2: Finding Complete Bipartite Subgraphs

TA handling this homework: Qianmu Yu qianmuyu@usc.edu

Due: March 1, Friday

100 points

In this homework, we consider the problem of finding complete bipartite subgraphs in a given bipartite graph. For example, graph $K_{2,3}$: ({1,2}, {a,b,c}) is such a subgraph in the following bipartite graph g. As shown in class, this problem may be reformulated as one of finding frequent itemsets, where the nodes in the left represent items and ones in the right baskets. Note that



Suppose the graph data is stored in a text file where each line represents an edge (starting with node in the left partition) in the graph. For example,

```
1,a
1,b
1,c
2,a
…
```

1. [60 points] Write a Python program “bipartite.py” that takes such a graph file and parameters (e.g., 2 and 3 as in $K_{2,3}$) of subgraph, and finds all required subgraphs in the graph (e.g., all $K_{2,3}$ subgraphs).

   Execution format:

   ```
   python bipartite.py <graph.txt> <s> <t>
   ```

   where <s> and <t> represent the number of nodes in the left and right part of the subgraph respectively.

   Your program should print the output, one subgraph per line as follows:

   ```
   {1,2}{a,b,c}
   …
   ```

Requirements: you must solve the program using one of the algorithms described in class for finding frequent itemsets.

2. [40 points] Implement the SON algorithm “son.py” in Spark for solving the same program in parallel. You are required to use mapPartitions(f) where the function f(p) should discover frequent itemsets in the partition p. You may reuse some of your codes in part 1 to implement f(p).