File: homework2.pdf

Author: P.J. Leyden

Date: October 1st, 2019

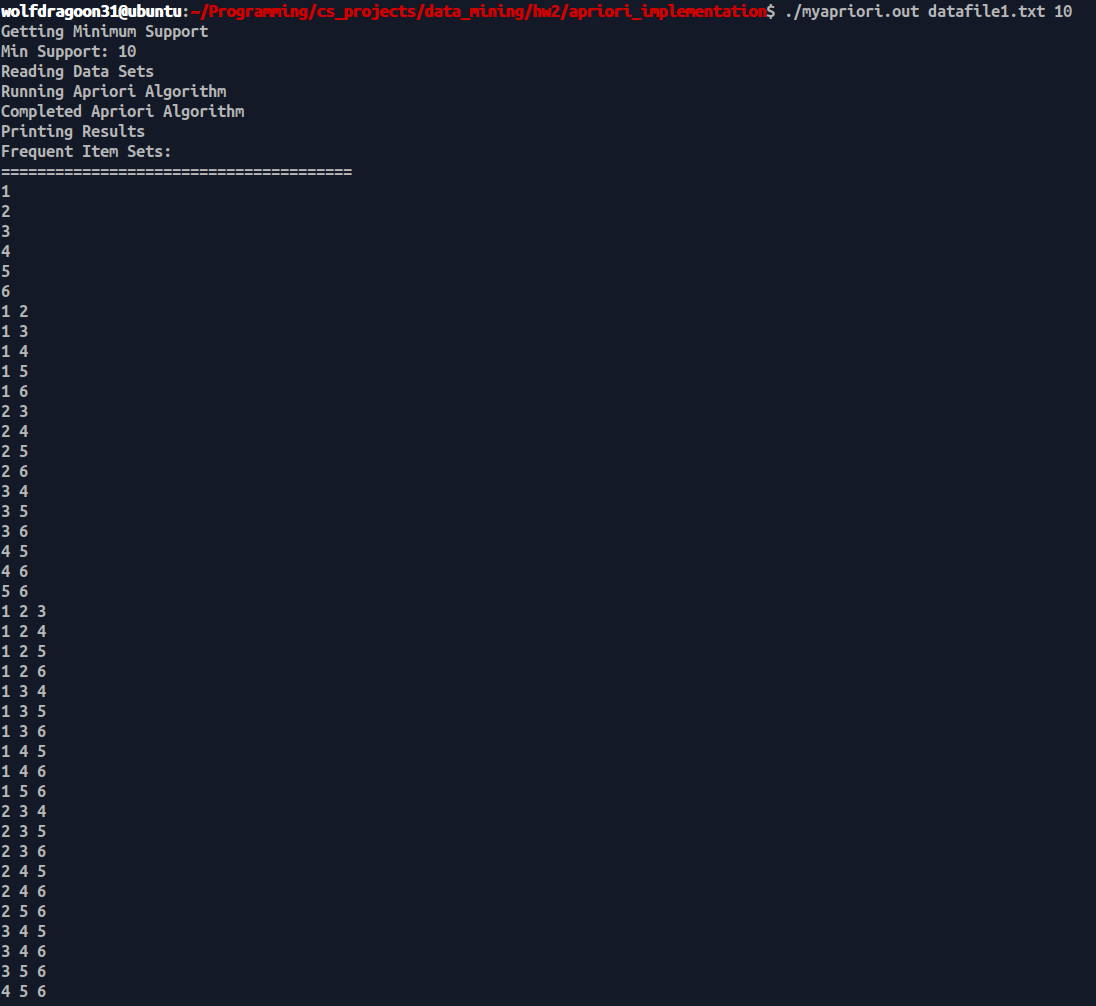
Homework 2

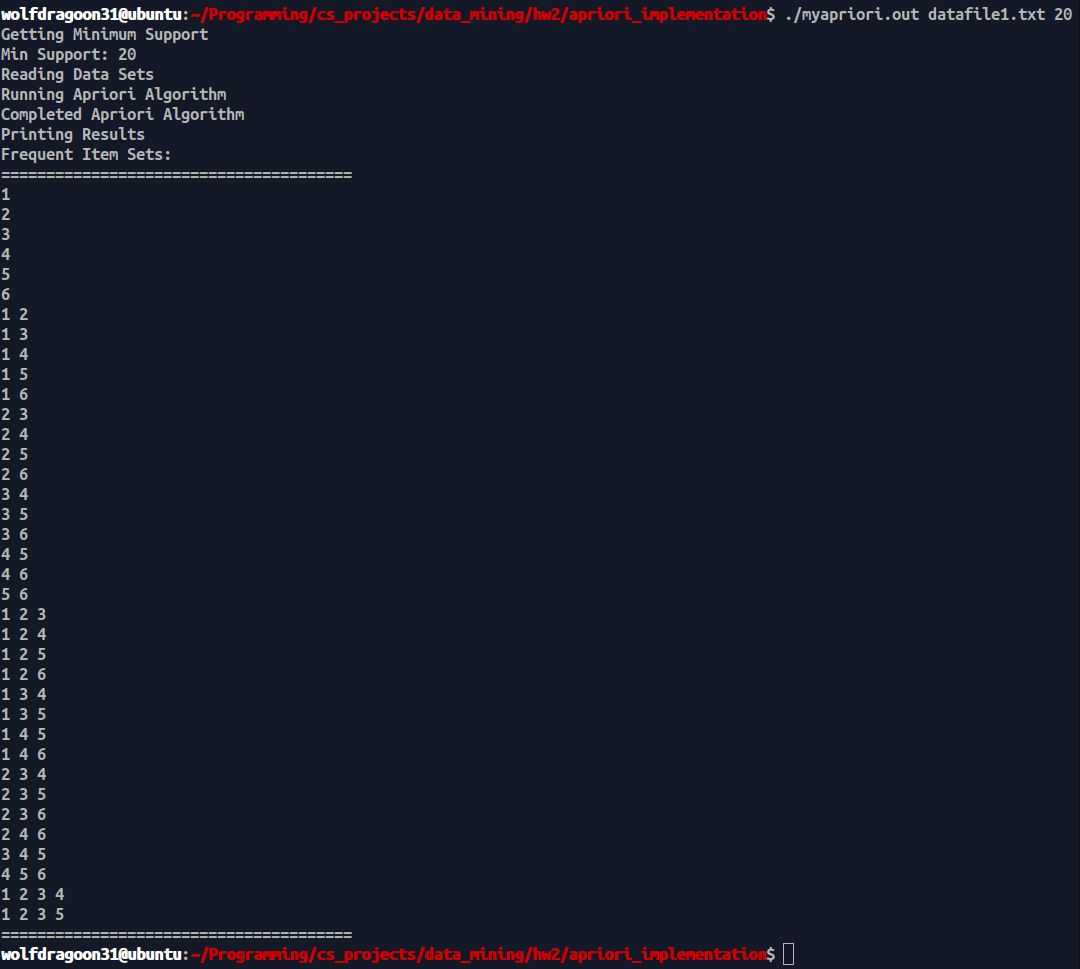
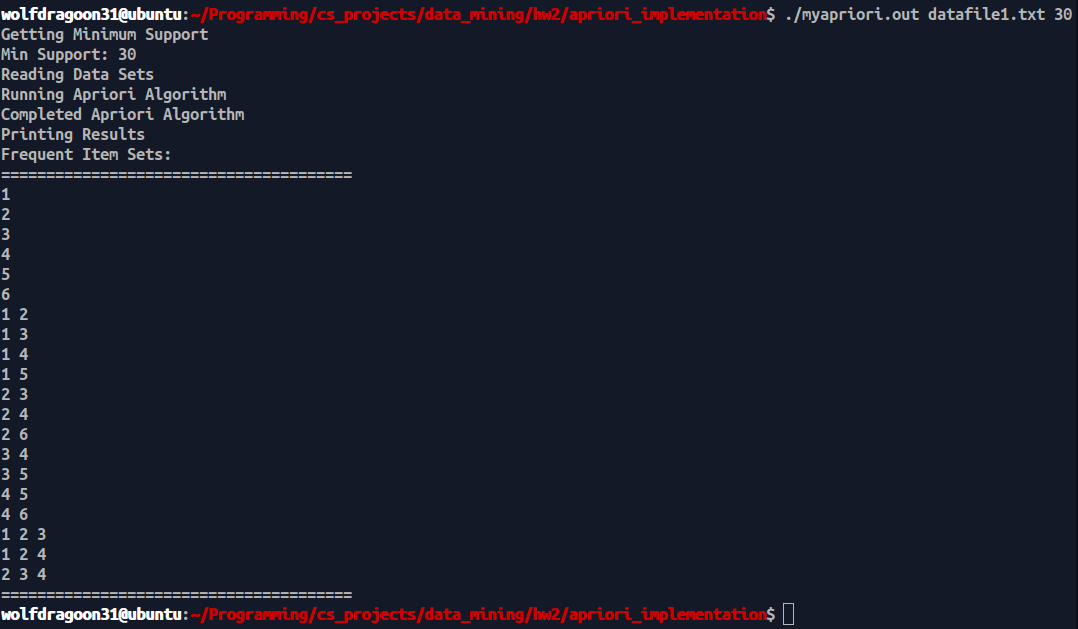
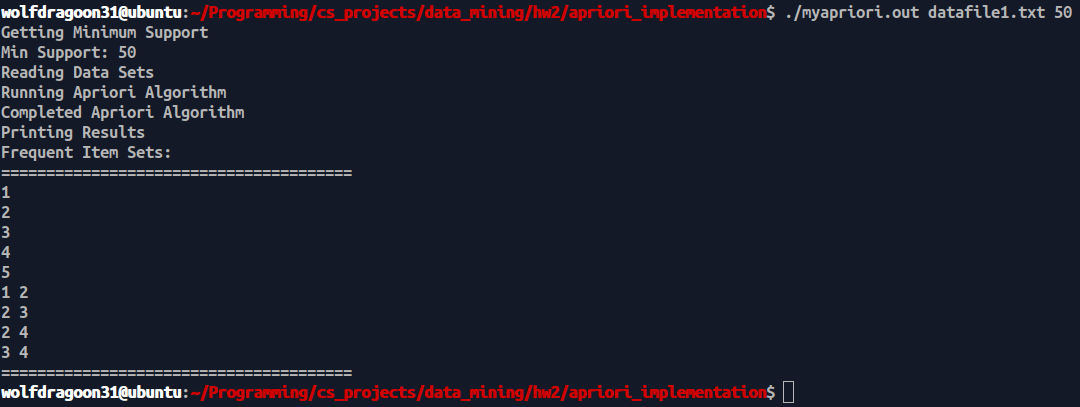
1. Code is in zip marked ‘myapriori.zip’. To run, use the ‘make’ command.
   1. The pseudo code:

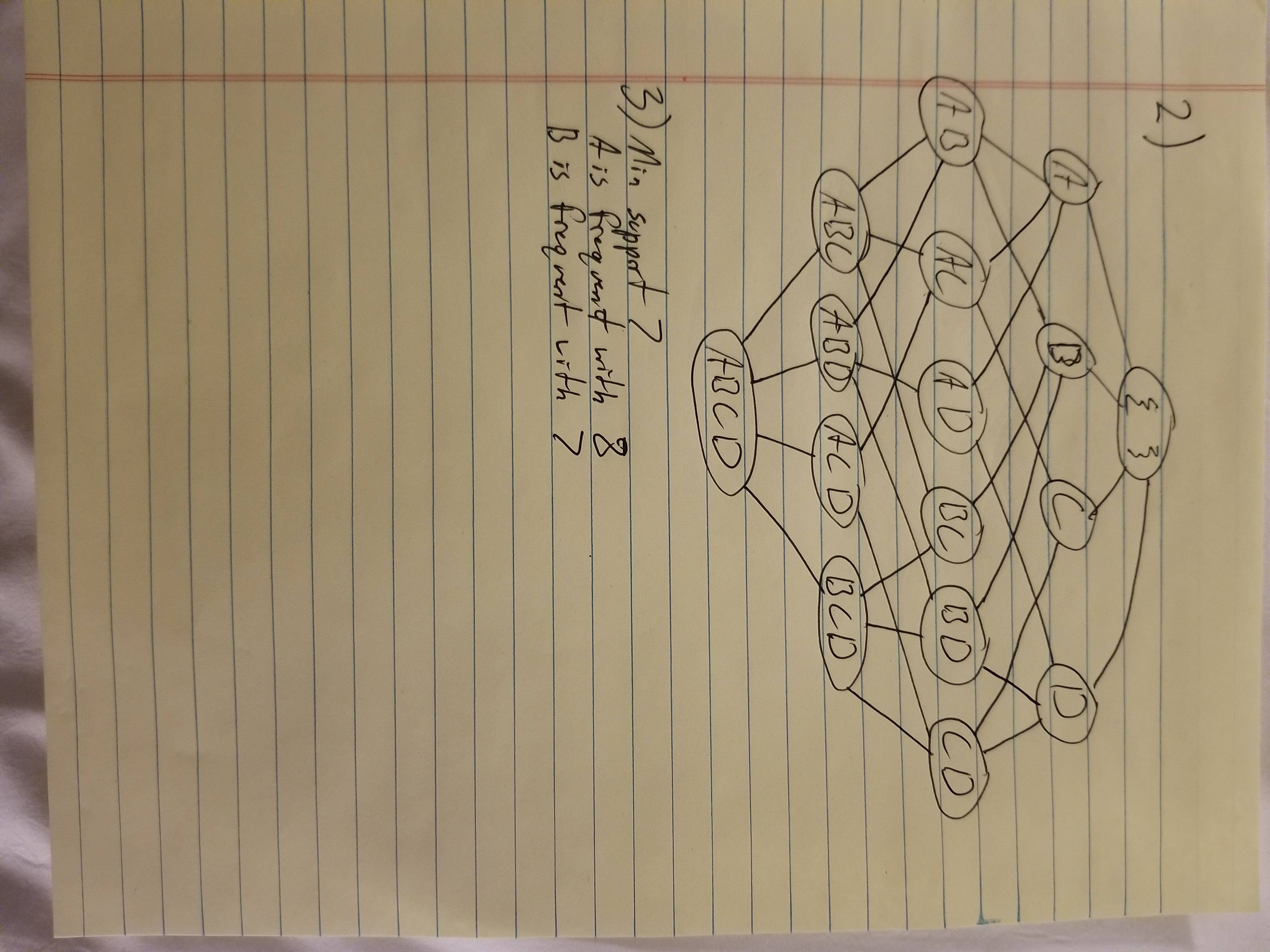
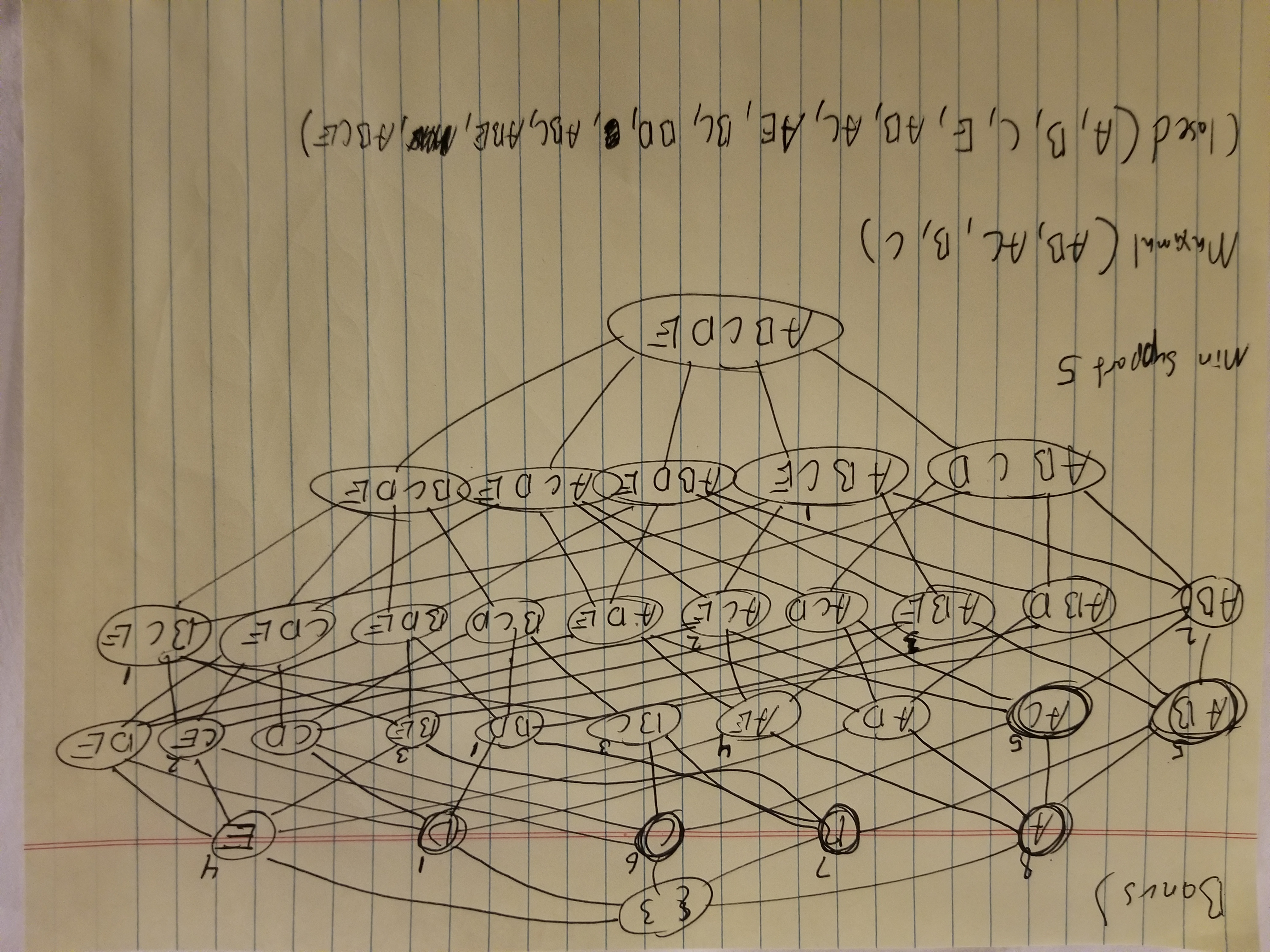
|  |
| --- |
| File: apriori\_pseudo\_code.txt  Author: P.J. Leyden  Date: October 1st, 2019  Apriori Algorithm Pseudo Code  //Preconditions  Let ck represent the series of subsequent candidate sets where k = length of the contained candidate sets.  Let fk represent the series of subsequent frequent sets where k = length of the contained candidate sets.  Let 'sets' represent the set of all itemsets  Let min\_sup represent the minimum support  Let results represent the final set ofall frequent item sets  //Generate the initial candidate set  for each set in sets  for each item in set  if item is unique  add to ck  max = ck.size  //main loop  for cur\_szie = 1; cur\_size <= max; ++cur\_size  //create map of frequency  map ck\_freq  for each set in c1  ck\_freq.push\_back(set, 0)  //iterate through the candidate set and check for frequency against database  for ck\_itr = ck.begin; ck\_itr != ck.end; ++ck\_itr  for each set in sets  i1 = ck\_itr->begin  i2 = set.begin  while i2 != set.end AND i1 != ck\_itr->end  if(i1 == i2)  ++i1  ++i2  if i1 == ck\_itr->end  ck\_freq.find(\*ck\_itr).second++  //determine the freqent sets and add them to the final result  for each pair in ck\_freq  if pair.second >= min\_sup  fk.add(pair.first)  //add frequent sets to results  for each set in fk  results.add(set)  //generate c(k+1)  ck.clear  ck = generate\_next\_candidate\_set(fk, f1)  //candidate gen function  set<set> generate\_next\_candidate\_set(f(k-1), f1)  //Preconditions  Let result be the resultant set of sets  Let cur\_set be the current set  //Algorithm  k = f(k-1).begin.size + 1  for each set in f(k-1)  for each element in f1  if set.find(element) == se.end  set\_clone = set  for each element2 in set\_clone  if element < element2  set\_clone.insert(element)  if result.find(set\_clone) == result.end  result.add(set\_clone)  break  return result |

Table 1 - Also in the file apriori\_pseudo\_code.txt in the zip file

* 1. Screen Captures of running code.
     1. Minimum Support 10%



* + 1. Minimum Support 20%
    2. Minimum Support 30%
    3. Minimum Support 50%

1. Question 2
2. Question 3
   1. With min support 7. The frequent item-sets are:
      1. {A} with support, 8
      2. {B} with support, 7
3. Answer to the Bonus Question: