The function returns us connected pairs and common pairs. In print step we first open file then we call flatMap over our function, then we call reduceByKey where we get total and current count, we filter only those pairs who have more than one friend. The first map maps the tuple ((user, friend), counts) to the tuple (user,(friend,counts)). The second map return us top N reccomendations.

We call this function by `friendsListRdd.lookup('ID')` this returns us dictionary of top N reccomendations.

Recomendations for 1:

- $924 \rightarrow 439, 2409, 6995, 11860, 15416, 43748, 45881$
- $8941 \rightarrow 8943, 8944, 8940$
- $8942 \rightarrow 8939, 8940, 8943, 8944$
- $9019 \rightarrow 9022, 317, 9023$
- $9020 \rightarrow 9021, 9016, 9017, 9022, 317, 9023$
- $9021 \rightarrow 9020, 9016, 9017, 9022, 317, 9023$
- $9022 \rightarrow 9019, 9020, 9021, 317, 9016, 9017, 9023$
- $9990 \rightarrow 13134, 13478, 13877, 34299, 34485, 34642, 37941$
- $9992 \rightarrow 9987, 9989, 35667, 9991$
- $9993 \rightarrow 9991, 13134, 13478, 13877, 34299, 34485, 34642, 37941$

Because it ingores $Pr(B)$. In some cases this may lead to incorect rules. For example occurence of B might be unrelated to A (`e.g.` A and B are independent, such that $conf(A \to B) = Pr\binom{B}{A} = Pr(B)$ but B have high support so that $A \to B$ is identified as a valid rule.

By observing the formulas we can see that `conv` and `lift` take $Pr(B)$ into account.

Lif is symetrical, confiedence and conviction are not since confidence and conviction are directional.

- $lift(A \to B) = lift(B \to A) = \frac{Pr(A,B)}{Pr(A)Pr(B)}$
- $conf(A \to B) = Pr(B|A)$ and $conf(B \to A) = Pr(A|B).Pr(A|B)$ and $Pr(B|A)$ might be different.
- $conv$ is based on $conf$ and is directional.

Example:

Let's have baskets AB, AC, AD then $S(A) = \frac{3}{3}$, $S(B) = \frac{1}{3}$ and $Pr(A, B) = \frac{1}{3}$. $conf(A \to B) = Pr(B|A) = conf(B \to A)$ since $\frac{\frac{1}{3}}{\frac{2}{3}} \neq \frac{\frac{1}{3}}{\frac{1}{3}}$

Similiraly $conv(A \to B) = \frac{1-S(B)}{1-conf(A \to B)} \neq \frac{1-S(A)}{1-conf(B \to A)} = conv(B \to A)$ since: $\frac{1-1/3}{1-1/2} = \frac{4}{3} \neq \frac{1-2/3}{1-1} = \infty$

Conviction and confiedence are desirable while lift is not. If B occurs every time then A occurs:

- $conf(A \to B) = 1$
- $conv(A \to B) \to \infty$
- $lif(A \to B)$ depends of the value $Pr(B)$ and may differ as B might occur in baskets which don't have A.

Example:

If we have baskets AB, AB, CD, EF, then $Pr(B|A) = 1, S(B) = \frac{1}{2}, Pr(D|C) = 1$ and $S(D) = \frac{1}{4}$. Then $lift(A \to B) = \frac{1}{\frac{1}{2}} = 2$ and $lift(C \to D) = \frac{1}{\frac{1}{4}} = 4$. Although both rules are 100% rules, they have different $lift$ scores.

Top 5 pairs of confiedence:

- DAI93865 $\rightarrow$ FRO40251 = 1.0
- GRO85051 $\rightarrow$ FRO40251 = 0.999176276771005
- GRO38636 $\rightarrow$ FRO40251 = 0.9906542056074766
- ELE12951 $\rightarrow$ FRO40251 = 0.9905660377358491
- DAI88079 $\rightarrow$ FRO40251 = 0.9867256637168141

Top 5 triples of confidence

- DAI23334, ELE92920 $\rightarrow$ DAI62779 = 1.0
- DAI31081, GRO85051 $\rightarrow$ FRO40251 = 1.0
- DAI55911, GRO85051 $\rightarrow$ FRO40251 = 1.0
- DAI62779, DAI88079 $\rightarrow$ FRO40251 = 1.0
- DAI75645, GRO85051 $\rightarrow$ FRO40251 = 1.0

The number of columns with m 1's out of n is $\binom{n}{m}$. Number of columns that have number 1 in one of k selected rows in $\binom{n-k}{m}$. Probability of number 1 in the chosen k rows is therefore divided by the former, so we get: $\frac{(n-k)m!(n-m)!}{m!(n-k-m)!n!}$. The $m!s$ cancel. We can rewrite equation to: $\binom{n-k}{n}\binom{n-k-1}{n-1}...\binom{n-k-m+1}{n-m+1}$ Each of m factors is at most $\binom{n-k}{n}$. When we calculate this expression to the end we get $\binom{n-k}{n}^m$ and this is their product at most.

We want $\frac{n-k}{n}^m \leq e^{-10}$. Equitably $(1-\frac{k}{n})^m \leq e^{-10}$. We can multipy and divide by $\frac{n}{k}$ which is then equivalent to $((1-\frac{k}{n})^{\frac{n}{k}})^{\frac{mk}{n}} \leq e^{10}$. We can assume $k < n$, then we can approximate $(1-\frac{k}{n})^{\frac{n}{k}}$ by $\frac{1}{e}$. When we do this we get desired condition $e^{-mk}n \leq e^{-10}$. Consequently, first exponent must less or equal to the second. Espcially $-\frac{mk}{n} \leq -10$, or $\frac{mk}{n} \geq 10$. Finally $k \geq 10\frac{n}{m}$. The lower bound on k is $10\frac{n}{m}$.

The two columns are $[0, 1, 0]^T$ and $[0, 1, 1]^T$. `Jaccard similiraty` is 0.5. If the cycle starts at either of the one of the first two rows, the minshash values are the same, while if the cycle starts at the last row, minhash values migh differ. The probability of the minhash values agreeing is $\frac{2}{3}$, when only cyclic permutations are allowed.

For each $1 \leq j \leq L$, each data point $xinT, Pr[x\varepsilon T \cap Wj] \leq p_2^k = \frac{1}{n}$ hence $E[|T \cap Wj|] \leq 1$. By linearatiy of expectation, $E[\sum_{j=1}^{L} |T \cap Wj|] \leq L$. Application of Markov's inequality gives us the desired probability bound.

$d(x^*, z) \leq \lambda$ for any $1 \leq j \leq L$ We have $Pr[Gj(x^* = Gj(z)] \geq p_1^k$ since $PrGj(x^* \neq Gj(z) \leq 1 - p_1^k = 1 - \frac{1}{L}$. Last equality by definition of k and L Independence of Gj we have $Pr[\forall 1 \leq j \leq L, Gj(x^* \neq Gj(z)] \leq (\frac{1-L}{L})^L \leq \frac{1}{e}$

U is the set of ANN points $U = x\varepsilon A; d(x,z) \leq c\lambda.x^*\varepsilon U.$. We have 2 ways a reported point is not an actual $(c,\lambda) - ANN$.

- None of the ANN points are hashedin the same buckets as z. $\forall 1 \leq j \leq L, Wj\cap = U = 0$ Denote E as the event : "none of the ANN points are hashed to the same buckets as z". Since $x^*\varepsilon U$ we have using question from b.
- There is at least one $(c,\lambda) - ANN$ point that is hashed to at least one of the buckets where z is hashed. If so there are also more than 3 L points at distance greater than c,λ in the union of those buckets. Although if there are less than 3 L points at that distance, the algorithm will return $a(c,\lambda) - ANN$. Denote F as event there are more than 3 L points at distance greater then c λ from z in the union of the buckets where z is hashed. In that case we can easily apply a question by $(a)^3$ and we know that F appers with a probabbility $< \frac{1}{3}$

If we name p the probability that the point returned by the algorighm is not a $(c,\lambda) - ANN$ we have $p = Pr[E \cup F] \leq Pr[E] + Pr[F] < \frac{1}{3} + \frac{1}{e}$ (union bound) algorithm always report actual $(c,\lambda) - ANN$ with probability $\geq 1 - \frac{1}{3} - \frac{1}{e}$

We need to note that we haven't shown that events F and E are independ so it would be incorrect to say $PR[E \cap F] \leq Pr[E] * Pr[F]$. But it is possible to define events similirar to those events and prove that they are independent and doing so would give a probability of correctness which is greater than the one we got with union bound.

Information sheet
CS246: Mining Massive Data Sets

Assignment Submission Fill in and include this information sheet with each of your assignments. This page should be the last page of your submission. Assignments are due at 11:59pm and are always due on a Thursday. All students (SCPD and non-SCPD) must submit their homework via Gradescope (http://www.gradescope.com). Students can typeset or scan their homework. Make sure that you answer each (sub-)question on a separate page. That is, one answer per page regardless of the answer length. Students also need to upload their code on Gradescope. Put all the code for a single question into a single file and upload it.

Late Homework Policy Each student will have a total of *two* late periods. *Homework are due on Thursdays at 11:59pm PT and one late period expires on the following Monday at 11:59pm PT*. Only one late period may be used for an assignment. Any homework received after 11:59pm PT on the Monday following the homework due date will receive no credit. Once these late periods are exhausted, any assignments turned in late will receive no credit.

Honor Code We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down their solutions independently, i.e., each student must understand the solution well enough in order to reconstruct it by him/herself. Students should clearly mention the names of all the other students who were part of their discussion group. Using code or solutions obtained from the web (GitHub/Google/previous year's solutions etc.) is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code very seriously and expect students to do the same.

Your name: _____

Email: _____ **SUID:** _____

Discussion Group: _____

I acknowledge and accept the Honor Code.

(Signed) _____