Sean Hinchee
Nicholas Losby
3/01/19

Lab 3

Experiment 1

The algorithm design for experiment 1 to pinpoint one-hop and two-hop relationships and allocate the respective credits involves two phases. The first phase re-emits the original data set, but emits as well a node pair of (b !a) for an existing reference (a b). The second phase then finds the two-hop relationship by identify itself as a valid two-hop middleman if there are flagged (prefixed with an '!') nodes. Credit for being a two-hop is then provided for every node for which there is not a pair (a !a), these are identified as self-references. Single-hop credit is obtained by manner of existing as a single-hop, no tracking is required. The credits are emitted on a (a, 1) pair and are summed and re-emitted to the final output as (a, sum). The top 10 can thusly be extracted.

Time complexity for communication: $O(n^2)$

Derivation: We iterate over every edge comparing to every other edge, worst case runtime for this is n^2, thus the time complexity for the whole algorithm is at worst, n^2.

Output:

Patent Number	Citations at one or two-hops
4503569	7281
4733665	6873
3868956	5487
4553545	5369
4655771	5311
4580568	5061
4512338	4937
3747120	4858
4445892	4716
4739762	4522

Experiment 2

The algorithm design for experiment 2 was largely based off the algorithm for experiment 1. The differences are with the initial emission of data as we now include the non-flagged version of the reversed input data. This allows us to check for a two-hop and a one-hop within the same mapreduce round to find triangles in the data set.Triplets are calculated by identifying all non-flagged elements (de-duplicated). The gcc is then calculated for each key and emitted, the gcc's are then summed and emitted for the final gcc.

Time complexity for communication: $O(n^2)$

Derivation: We recycled most of the previous algorithm while maintaining the same for loop, thus the time complexity for the whole algorithm is at worst, n^2.

Output:

```
sum-gcc 7.8891391E7
```