CSC111 Assignment 3: Graphs, Recommender Systems, and Clustering

Zixiu Meng, Jingyang Yu

March 27, 2021

Part 1: The book review graph and simple recommendations

1. Complete this part in the provided `a3_part1.py` starter file. Do **not** include your solution in this file.

2. 1. the first for loop has a running time of $\Theta(m)$ because the loop iterate m times and the opening of csv file and each iteration of the loop body is a constant time operation.
   2. the second for loop has a running time of $\Theta(n)$ because the loop iterate n times. though the loop body contains two steps, addedge and addvertex are both constant time operation.
   3. hence the overall time complexity of the implementation is $\Theta(m + n)$

3. Complete this part in the provided `a3_part1.py` starter file. Do **not** include your solution in this file.

4. Complete this part in the provided `a3_part1.py` starter file. Do **not** include your solution in this file.

Part 2: Weighted graphs, recommendations, review prediction

Complete this part in the provided `a3_part2_recommendations.py` and `a3_part2_predictions.py` starter files. Do **not** include your solution in this file.

Part 3: Finding book clusters

1. Complete this part in the provided `a3_part3.py` starter file. Do **not** include your solution in this file.

2. Complete this part in the provided `a3_part3.py` starter file. Do **not** include your solution in this file.

3. (a) 1.the outer for loop iterates m_1 times.
   2.the inner for loop iterates m_2 times.
   3.the inner loop body is a constant time operation.
   4.the overall time complexity is $\Theta(m_1 m_2)$

   (b) 1. "if score is greater than best" part is a constant time operation.
   2. "cross cluster weight(graph, c1, c2)" has a running time of $\Theta(c_1 c_2)$
   3. since cluster1 has fixed size k, the running time of "cross cluster weight(graph, c1, c2)" is $\Theta(c_2)$
   4. the total running time is $\sum_{i=1}^{j} ac_{2i}$ where j is the number of iteration
   5. the sum of all cluster size is n which means $\sum_{i=1}^{j} c_{2i}$ = n. Therefore, the total running time is kn.
   6. k can be written as n/p for some number p. 5.hence the running time for inner loop is n*n/p, which$\mathcal{O}(n^2)$

   (c) 1. the set operation takes n times
   Inside the for loop:
   2. the first two reassignment takes constant time.
   3. as mentioned previously, the inner loop has an upper asymptote of n^2
   4. set.update takes the proportion of size of best_c1, which means it takes at most n steps.
   5. list.remove takes at most n times
   6. the loop body of outer loop takes $n^2 + 2n + 1$times, which is $\mathcal{O}(n^2)$
   7. the outer loop iterates $n - k$ times
   8. the overall upper asymptote for this algorithm is $\mathcal{O}(n^2(n-k)+n)$, which is $\mathcal{O}(n^2(n-k))$

   (d) *Not to be handed in.*