Summer of bitcoin Submission

Table of content

Reasoning behind logic

Went with Fractional knapsack (greedy) approach to the problem statement

- First made the CSV file readable to C++ language
- Created a map of the transactions, key being Index of transaction
- Created a "set" of all transactions, with parameters, Ratio = fee/weight, and mapped key of the transaction as the second parameter.
- Once obtained transaction with highest ratio->fee, checked if the parents of the transaction exist before jotting down the transaction-id.
- If all flags came in favor of the transaction, added the transaction to the final block.

Tradeoffs

1. With the implementation of DSU(Disjoint Union Graphs), the complexity could have been reduced to O(n^2) instead of currently being O(n^3). Implementation couldn’t be done due to lack of proper knowledge/time constraints for implementation.

2. Combing all transactions having parent transactions into nodes and thus working over nodes could have reduced time complexity. This can be done when dealing with datasets where every transaction has a parent transaction.

Complexity of Code O(n^3)
Total weight of the final Block: 3999936
Total fee earned by miner **(objective):** 4803570
Total transactions considered: 2668