Operating Systems

Assignment-3

Group Members

Atishay Jain	20CS30008
Gaurav Malakar	20CS10029
Monish Natrajan	20CS30033
Roopak Priydarshi	20CS30042

- We have used multi-source Dijkastra algorithm in each consumer process with the n/10 nodes as sources.
- We have used 5 shared memories to facilitate communication between the processes as follows:
- graph: to store the graph as adjcency list (assuming max list size)
- probs: helper data structure to calculate the probability
- slots: to store the source nodes for each consumer process
- dists: to store distance array of each consumer process
- prnts: to store the parent array for each consumer process

Optimized part:-

- For optimizing our program we used shared memory to strore the distance array and parent array for each segment.
- Accessing the huge shared memory is an overhead here.
- This results in a slower executin in first call of the consumers but a clear decrease in execution time is observed in the later calls.
- The main idea is to use the previously calculated distance array to avoid recalculations of these distances.
- This saves time as we can see in the results below:
- When each consumer process with n/10 has been called atleast once, the updatation time will be less as we can see below:

Iteration No.	Unoptimised	Optimised
1	0.584359	0.665854
2	0.587487	0.413464

3	0.904302	0.854838
4	0.903144	0.637643
5	0.596323	0.443366
6	0.912137	0.659016
7	0.604499	0.536342
8	0.607112	0.537243
9	0.601311	0.594729