**Description :**

When new nodes are added by the producer process, the shortest paths between such new nodes and other nodes need to be computed, and the shortest paths between old nodes might need to be updated.

Let, V = 4000, E = 8 \* 104 , M (No. of new nodes) = 30.

**Ordinary Approach :**

Apply Dijkstra’s algorithm on all nodes after the new nodes have been added.

Time complexity = (V+M) \* E \* logV = 1.16 \* 109 operations

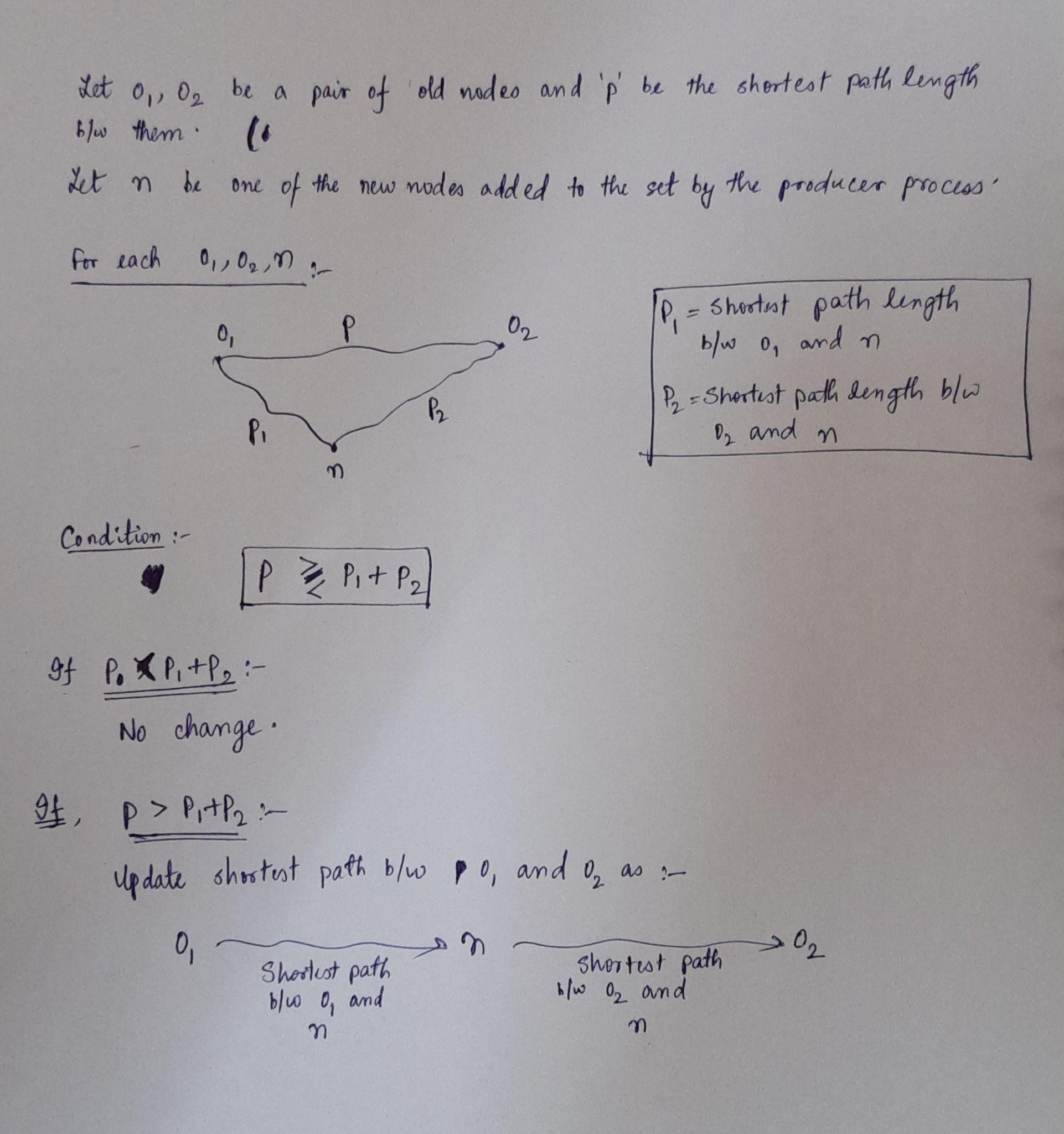
**Our Proposed Approach :**

1. We first apply Dijkstra’s algorithm on the new nodes to compute the shortest paths between :

* pairs of new nodes,
* pairs of old and new nodes.

which gives a total time complexity of = M \* E log V = 8.64 \* 106 operations.

1. Now, for updating the shortest paths between pairs of old nodes (if needed), we apply the following optimization :

****

**Correctness of the Approach :**

The approach can be validated by the following statements about the optimization step :

* To perform the optimization step as described above, the shortest paths between new nodes and other nodes need to be precomputed, which is done in Step 1 of our approach.
* If the introduction of new nodes still doesn’t create a shorter path between two old nodes, then the shortest paths between these nodes remains the same. Otherwise, the previously shortest path between two old nodes is replaced by a new shortest path, involving one or more new nodes. Hence, this approach is guaranteed to give the shortest possible path between all pairs of nodes.
* Though this approach seemingly involves only one new node during shortest path finding, multiple new nodes can get involved in the shortest path implicitly because the shortest paths between (o1 and n) and (o2 and n) might involve new nodes.

**Total Time Complexity :**

For the above optimization step,

* Total no. of pairs involving old nodes = V2 / 2
* Total no. of new nodes = M

Thus, total time complexity of the above pairwise optimization step + applying Dijkstra on all new nodes gives :

= M \* V2 / 2 + M \* E log V

= 0.248 \* 109 operations

**Improvement = 1.16 / 0.248 = 4.65**