Assignment No.: B6

Roll No.

- **Title :**
  To implement concurrent prims algorithm using OPENMP.

- **Problem Definition :**

  Implement concurrent prims algorithm to find shortest distance using OPENMP.

- **Learning Objective :**

  To study the concurrent prims algorithm using OPENMP.

- **Learning Outcome :**

  Successfully implemented prims algorithm using OPENMP.

- **Software and Hardware Requirement:**

  - Latest version of 64 Bit Operating Systems Open Source Ubuntu 14.04
  - Multicore CPU equivalent to Intel i5/7 4^{th} generation
  - OPENMP installed on the operating system.

- **Theory :**

  1. OpenMP :

OpenMP consists of a set of compiler #pragmas that control how the program works. The pragmas are designed so that even if the compiler does not support them, the program will still yield correct behavior, but without any parallelism.

2. Prim's Algorithm :

Prim's algorithm is a greedy algorithm that finds a minimum spanning tree for a connected weighted undirected graph. It finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized.This algorithm is directly based on the MST(minimum spanning tree) property.

3. Example of Prims Algorithm:

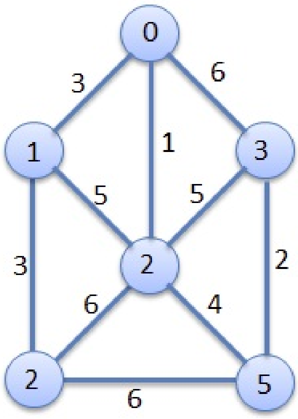


Figure 1: A simple Weighted Graph

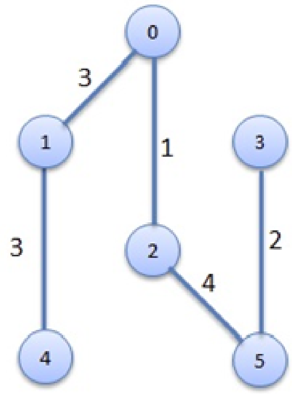
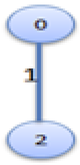


Figure 2: Minimum-cost Spanning Tree

Procedure for finding Minimum Spanning Tree

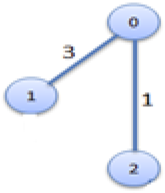
Step 1:

No. of Nodes	0	1	2	3	4	5
Distance	0	3	1	6	∞	∞
Distance from		0	0	0		



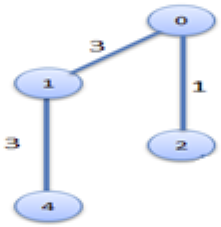
Step 2:

No. of Nodes	0	1	2	3	4	5
Distance	0	3	0	5	6	4
Distance from	0		2	2	2	



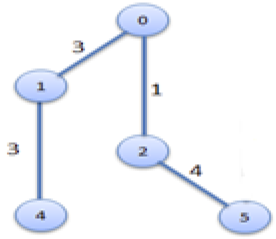
Step 3:

No. of Nodes	0	1	2	3	4	5
Distance	0	0	0	5	3	4
Distance from			2	1	2	



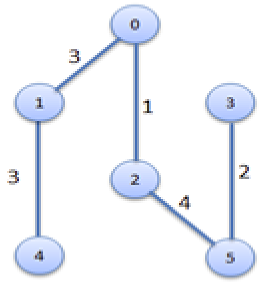
Step 4:

No. of Nodes	0	1	2	3	4	5
Distance	0	0	0	5	0	4
Distance from				2	2	



Step 5:

No. of Nodes	0	1	2	3	4	5
Distance	0	0	0	3	0	0
Distance from				2	2	



Minimum Cost = 1+2+3+3+4 = 13

4. Algorithm:

1. Declare the variables and set the minimum distance to 1000
2. Accept the number of nodes and weights among nodes from user.
3. To initialize the data call initialization() i.e go to step 10.
4. Calculate for all the nodes.
5. For every node in minimum spanning tree.
6. Declare OpenMP's directive with the appropriate scheduling.
7. Find the minimum weight.
8. Add the local min_distance to the total_min_distance.
9. Add the next node in the U set.
10. Subtract the elements of the column in which the new node is assosiated with.
11. Increment the nodes that are in the MST.Go to step4 till all nodes.
12. Go to step 14.
13. Initialize the U-set.
14. Store the first node into the U set.
15. Delete the first node.
16. Increment by one the number of nodes that are inside the U set.
17. Print all the nodes in MST in the way that they stored in the U set and display the minimum cost of spanning tree.
18. Exit.

5. Flowcharts:

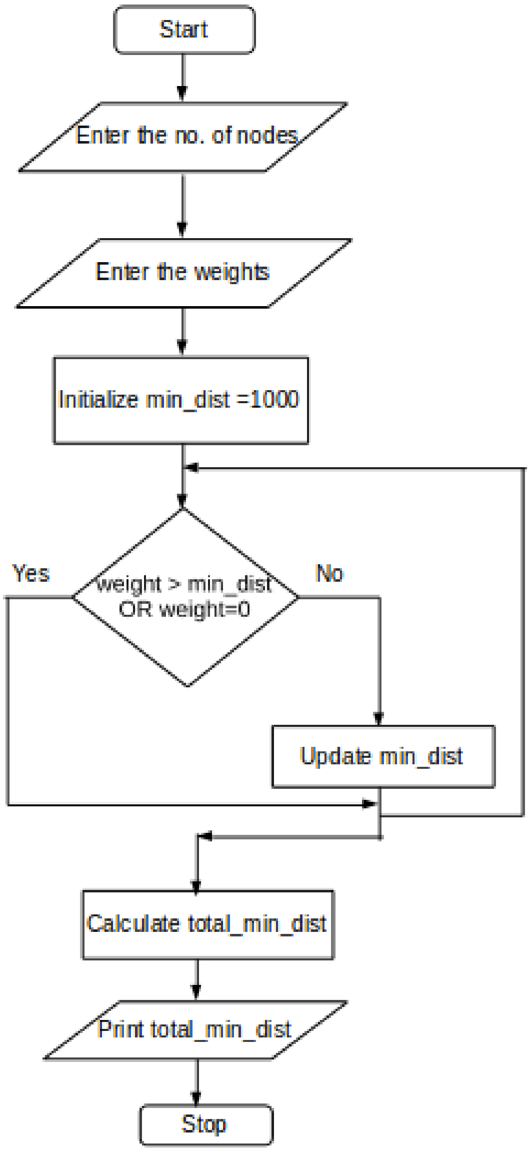


Figure 3: Flow chart

6. Mathematical model:

Aim:

Let system 'S' be the solution for the calculation of concurrent prim's algorithm.

Mathematical model using Set theory:

$$Squareof2digitnumber(S)$$
$$L = \{(s, e, i, o, f, DD, NDD, success, failure)\}$$

s= Initial state
e= End of state
i=input set.
o=output set.
DD=deterministic data
NDD=Non deterministic data
success=system reaches desired state
failure=desired outcome not generated.
i={i_1,i_2}
i_1:Number of nodes
i_2:Cost of edges
o=minimum cost of spanning tree
f={initialization(),delete_elements()}
initialization()={Initializes the vertex set and points to first node}
delete_elements()={deletes the specified element} Success:

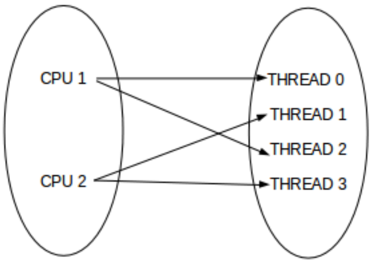
- Accurate output is generated.
- Successful termination of the program.

Failure:
System will fail under following conditions:

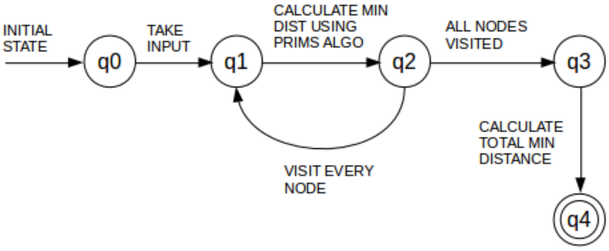
- Incorrect,inconsistent input from user.
- Syntactical and logical errors.
- Desired output is not generated.

We have implemented two classes.
CPU class is used to initialize the variables, accept input and create threads.
Thread class is used for executing the main algorithm concurrently.

Venn diagram:



State diagram:

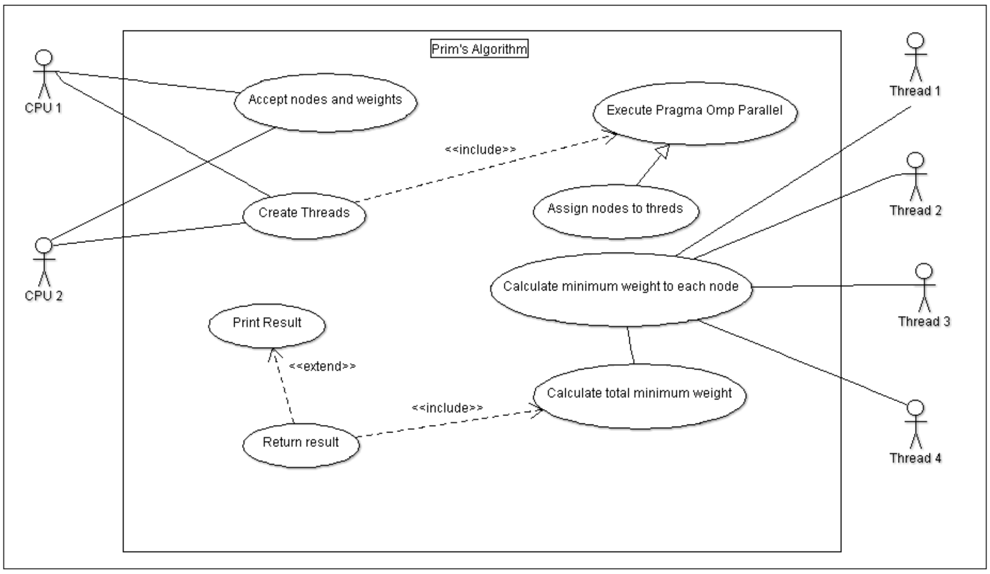


7. SRS:

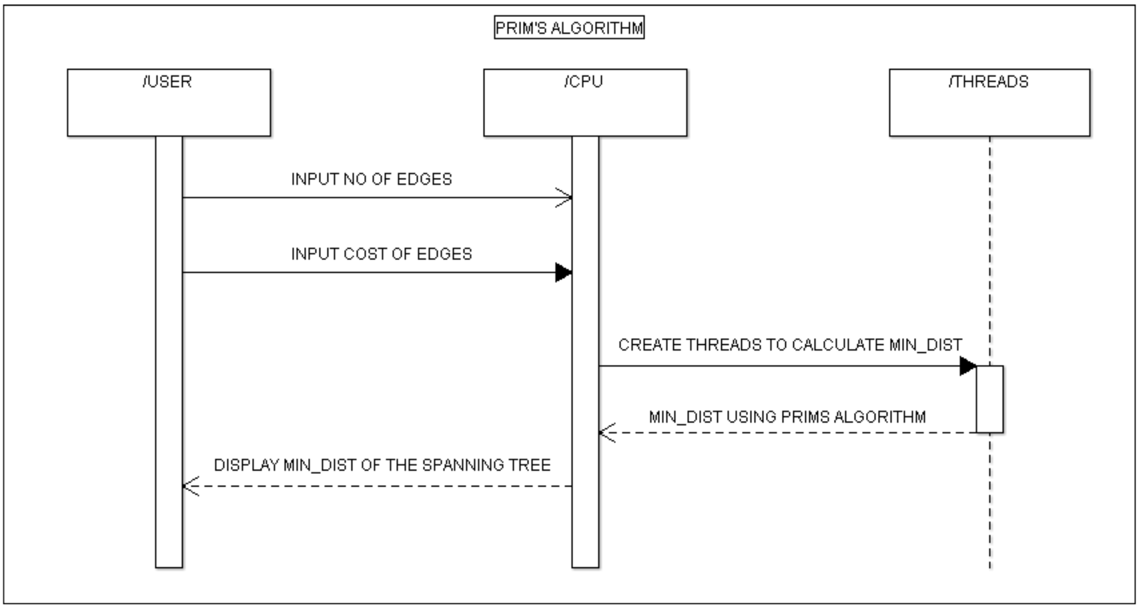
Software Specification	Description
1.Project Scope	To design a program that calculates minimum distance using Prim's Algorithm.
2.Functional requirements	Input given must be valid.
3.Non-functional requirements	Quality, Usability, Performance.
4.Design	The cost of edges must be an integer value and should be accurate.
5.Implementation	Using OPENMP thread concept.
6.Hardware Requirements	Dual core CPU i.e. 4 threads.
7.External Interfacing	Not required.

8. UML Diagrams:

Use case Diagram:

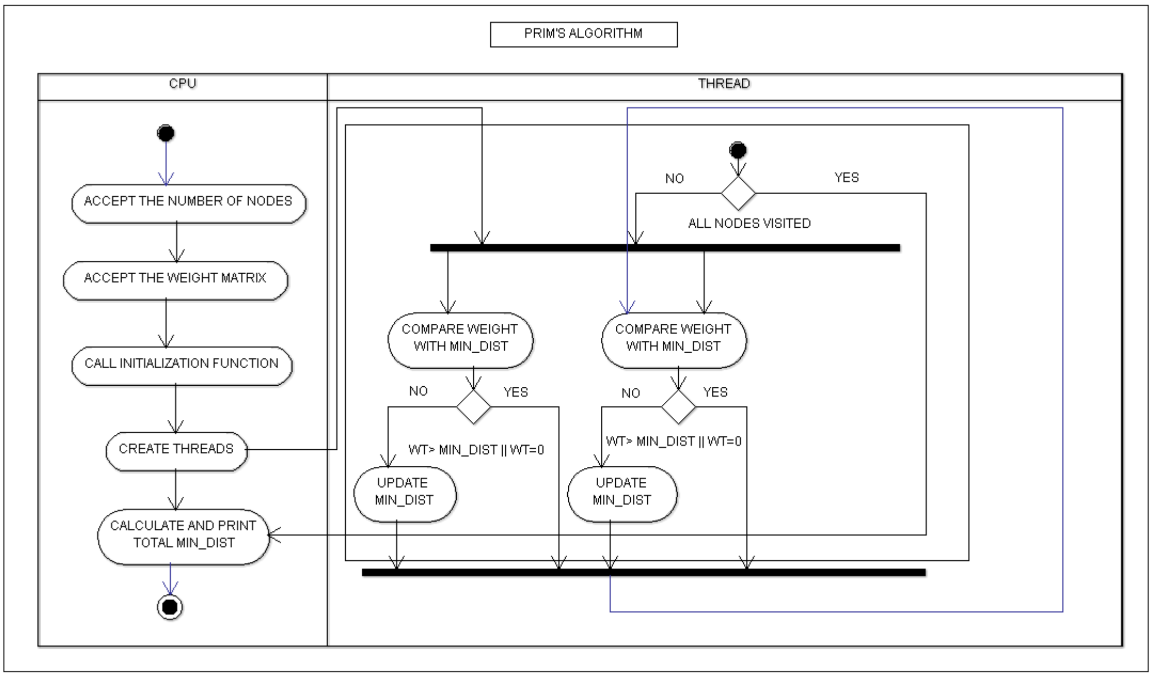


Sequence Diagram:

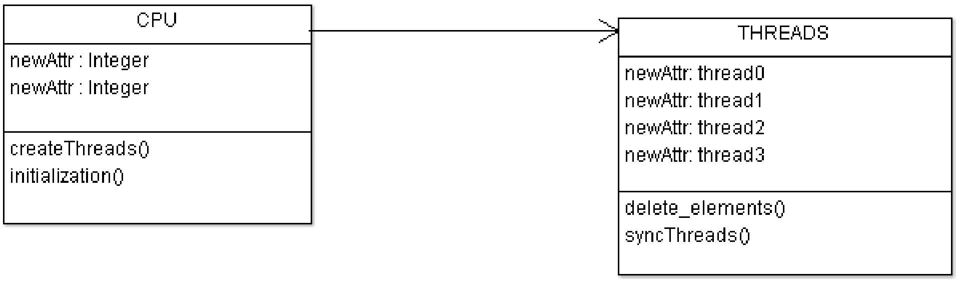


PRIM'S ALGORITHM
/USER
/CPU
/THREADS
INPUT NO OF EDGES
INPUT COST OF EDGES
CREATE THREADS TO CALCULATE MIN_DIST
MIN_DIST USING PRIMS ALGORITHM
DISPLAY MIN_DIST OF THE SPANNING TREE

Activity Diagram:



Class Diagram:



- **Input:**

The input is given as the weight of each node to every other node in the spanning tree.

- **Output:**

The output is the minimum cost of the spanning tree given as input.

- **Conclusion :**

  Hence, we have successfully implemented concurrent Prim's algorithm using OPENMP.

Course Outcomes	Achieved Outcome
CO I : Ability to perform multi-core, Concurrent and Distributed Programming.	√
CO II : Ability to perform Embedded Operating Systems Programming using Beaglebone	
CO III :Ability to write Software Engineering Document.	√
CO IV :Ability to perform Concurrent Programming using GPU.	

FAQ:

- What is Prim's Algorithm? What are its advantages?
- Compare Prim's algorithm with other MST algorithms.
- Explain in detail about OPENMP.
- What do you mean by minimum spanning tree?
- How to achieve concurrency in Prim's algorithm?
- What are various applications of Prim's algorithm?
- Why do we use concurrency in programming?