Prim's Algorithm

[redacted]

[redacted] Minimum Spanning Tree (MST) [redacted] **Prim's Algorithm** [redacted] MST

[redacted] connected [redacted] edges [redacted] `vertices - 1` [redacted] Minimum Spanning Tree [redacted]

1. [redacted] Graph

- [redacted] `inf` [redacted] **weight** [redacted] vertex
- [redacted] **Matrix** [redacted] **Graph**

2. [redacted] `addEdge()`

- [redacted] (edge) [redacted] vertex [redacted] weight [redacted] Matrix [redacted] vertex [redacted]

3. [redacted] `prim()`

- [redacted] `prim()` [redacted] Minimum Spanning Tree [redacted]:
  - [redacted] vertex 0
  - [redacted] MST [redacted]
  - [redacted] MST [redacted] MST [redacted] vertex [redacted] MST
  - [redacted] MST [redacted]

4. [redacted]

- [redacted] vertices, edges, [redacted]

5. [redacted]

- [redacted] Minimum Cost [redacted] MST [redacted]

Pseudocode

```
[redacted] prim([redacted], [redacted]):
    [redacted] = [redacted]
    [redacted][0] = [redacted]  // [redacted] 0

    [redacted] = 0

    [redacted] i [redacted] 0 [redacted] - 1:
        [redacted] = [redacted]
        [redacted] = -1

        [redacted] v [redacted] 0 [redacted] - 1:
            [redacted][v] [redacted][i][v] < [redacted]:
                [redacted] = [redacted][i][v]
                [redacted] = v

        [redacted] -1:
            [redacted][[redacted]] = [redacted]
            [redacted] += [redacted]

    [redacted]

[redacted] main():
    [redacted] "===== [redacted] Prim ====="

    [redacted] = [redacted]("[redacted]: ")
    [redacted] = [redacted]("[redacted]: ")

    [redacted] < 1 [redacted] < 1:
        [redacted] "[redacted]"
        [redacted]

    Graph.init([redacted])

    [redacted] i [redacted] 0 [redacted] - 1:
        u = [redacted]("[redacted] " + (i + 1) + ": ")
        v = [redacted]("[redacted] " + (i + 1) + ": ")
        w = [redacted]("[redacted] " + (i + 1) + ": ")

        u -= 1
        v -= 1

        [redacted] u < 0 [redacted] u >= [redacted] v < 0 [redacted] v >= [redacted]:
            [redacted] "[redacted]"
            [redacted]

        [redacted] w <= 0:
            [redacted] "[redacted]"
            [redacted]

        Graph.addEdge(u, v, w)

    [redacted] < [redacted] - 1:
        [redacted] "[redacted]"
        [redacted]

    [redacted] = prim(Graph.matrix, [redacted])

    [redacted] "[redacted]: " + [redacted]
```

[redacted]

---

1. [redacted] Command Prompt ([redacted] Terminal) [redacted] Path [redacted] `prim.go`
2. [redacted] `go run prim.go` [redacted] Compile [redacted] Run Program
3. [redacted]

4. [redacted]