**Dijkstra Algorithm**

Team (24)

Participant:

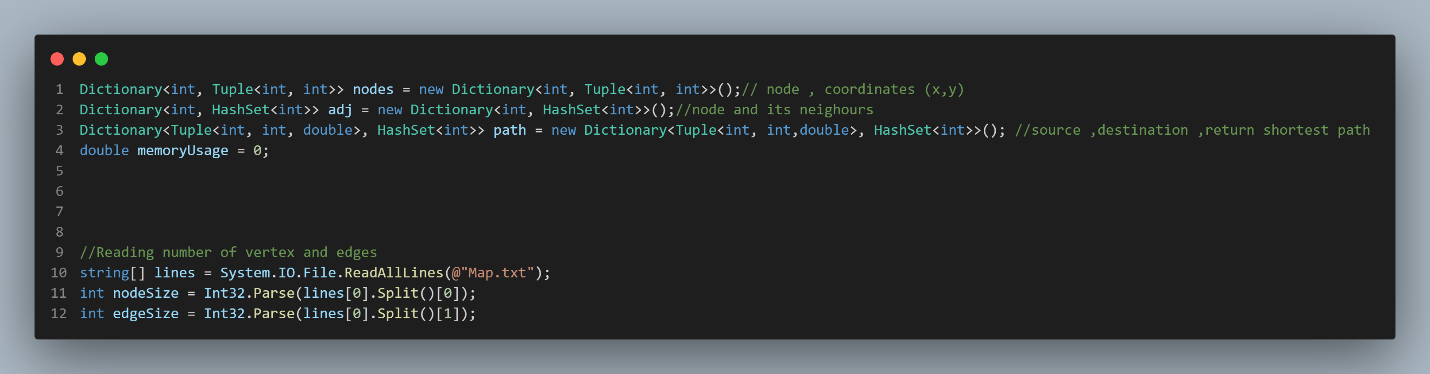
* Ahmed Emam Mohamed Emam .(Sec 1)
* Ahmed Mohamed Hussien Ahmed .(Sec 1)

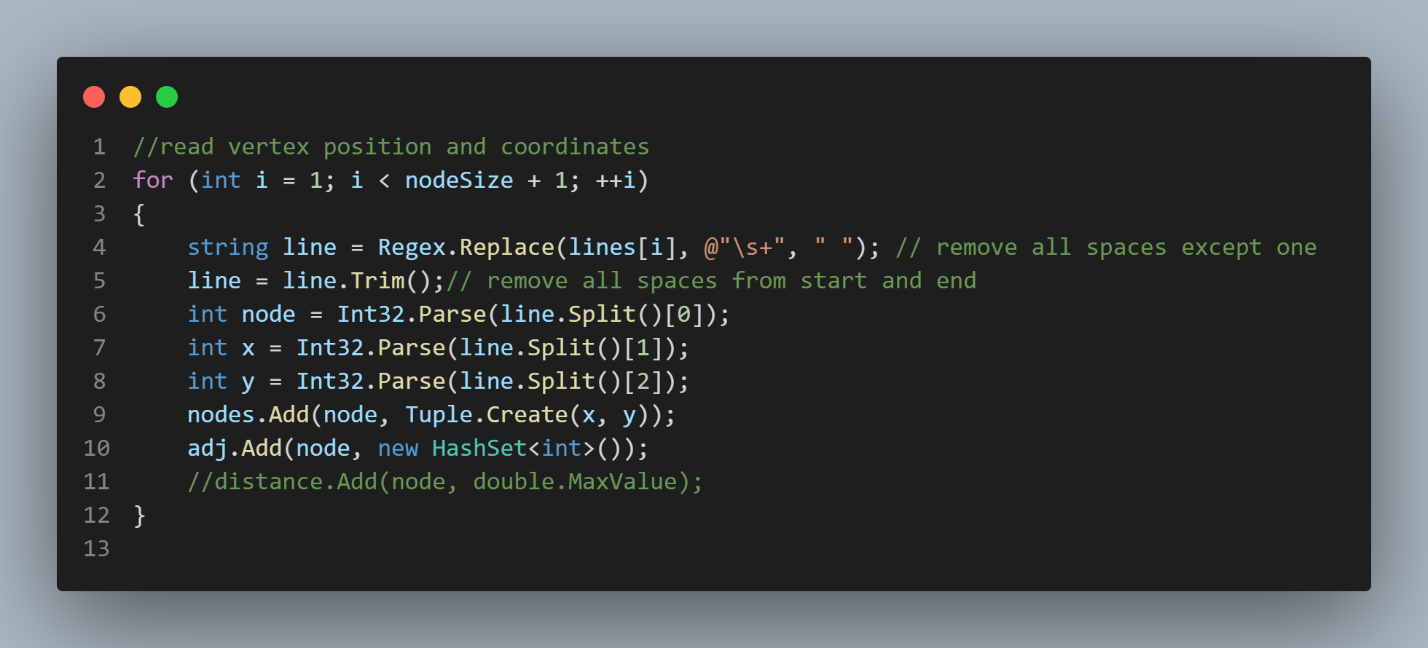
**Lecturer :**

**Dr . Ahmed Salah**

Dijkstra Algorithm Code Analysis :

1. **Declaration.**

Declaration O(1).

1. **Reading Vertices From Map File:**

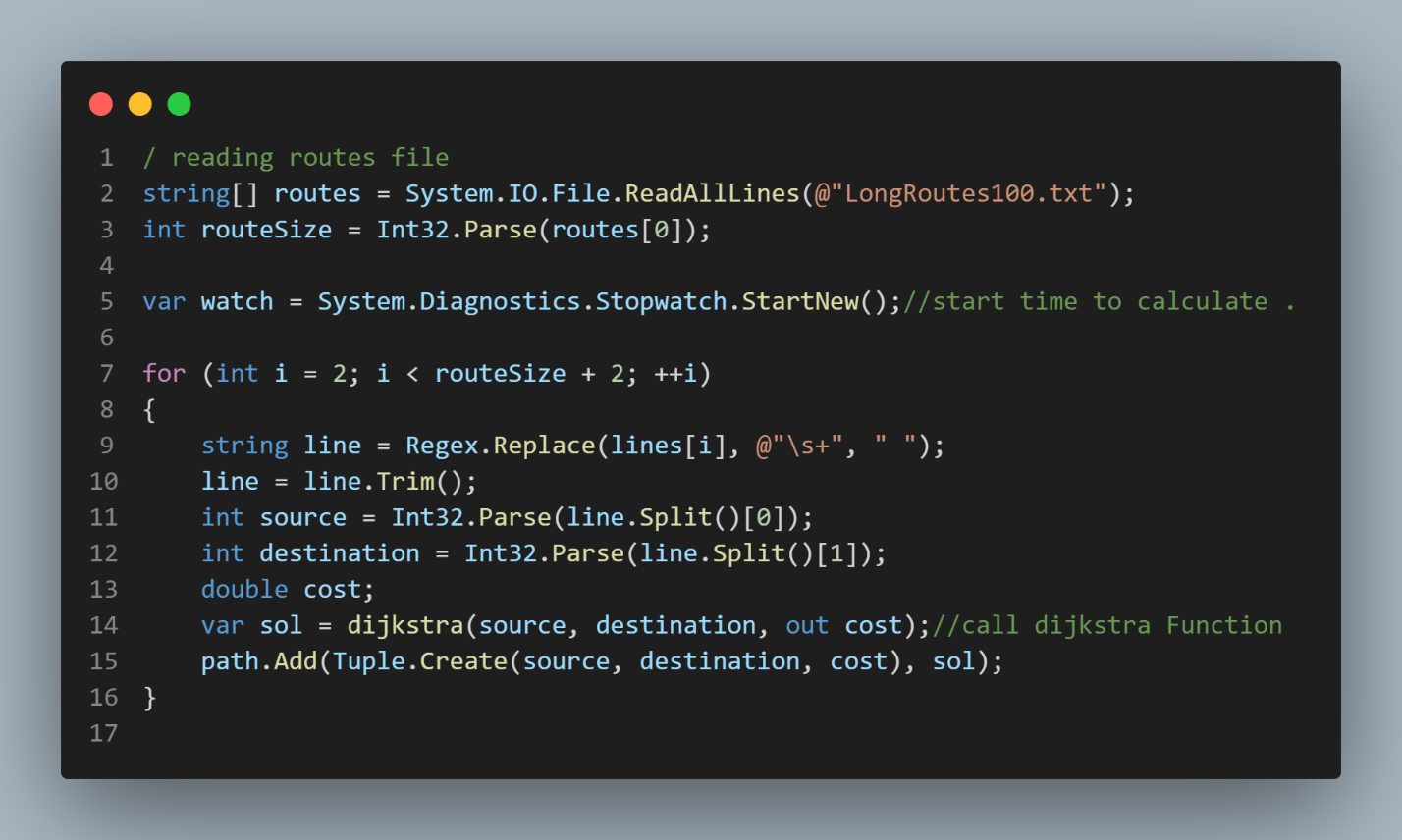
Reading Loop Θ(V) V is Number of Vertices.

1. **Reading Edges From Map File.**

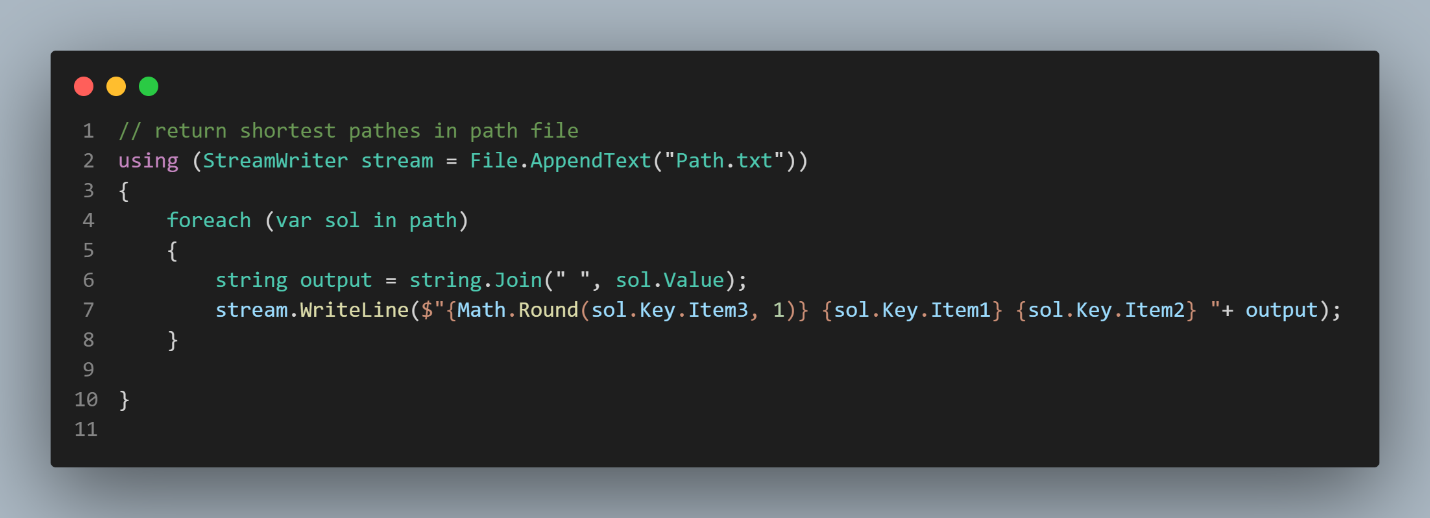


Reading Loop Θ(E) E is Number Of Edges**.**

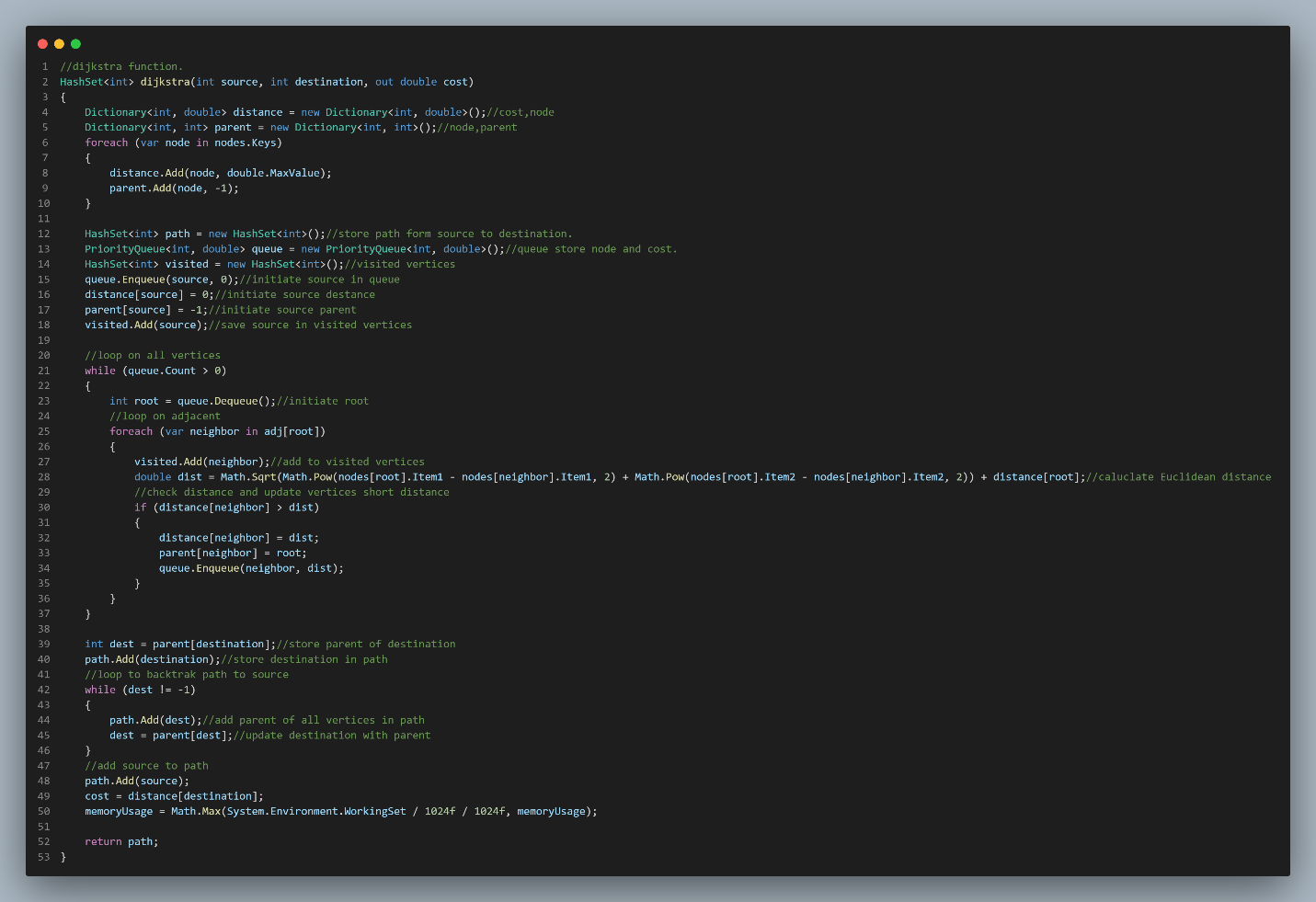
1. **Reading Source And Destination From Route File.**

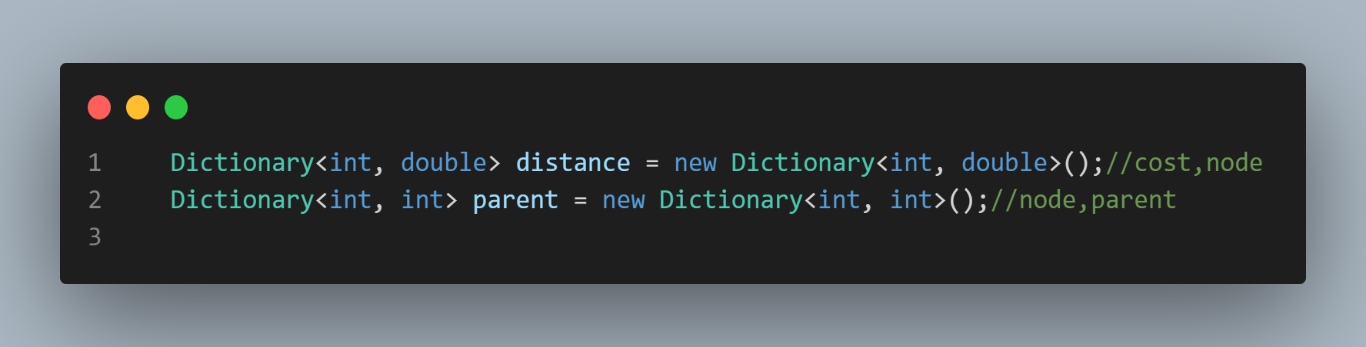
****Reading Loop O(R +V ) R Is Number of Routes .

1. **Print Shortest Paths of Routes:**

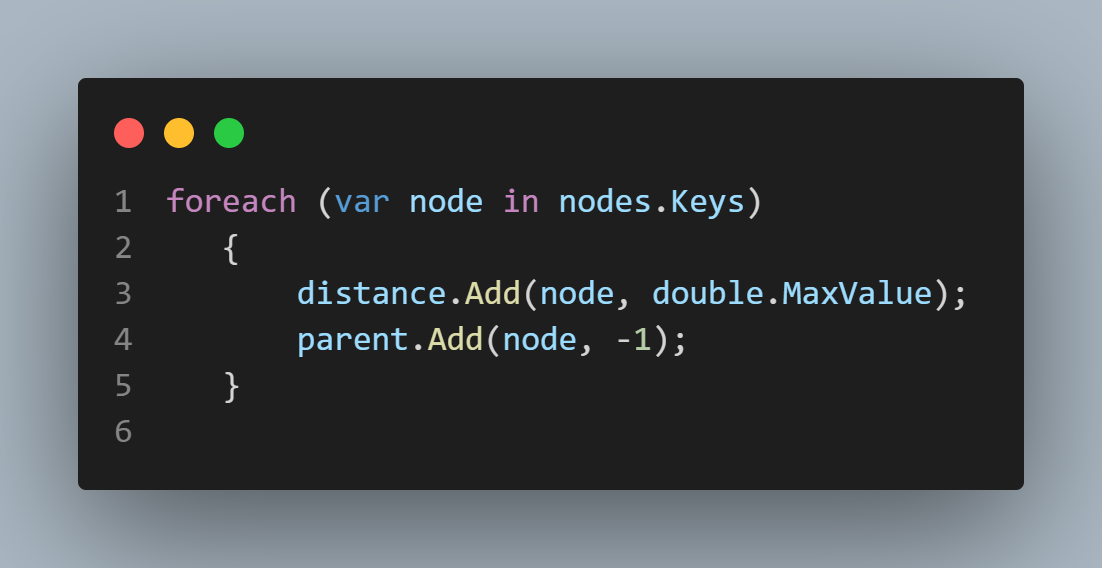
Print O(R) R Is Number Of Routes.

1. **Dijkstra Function:**

 Dijkstra Function O( E ) .

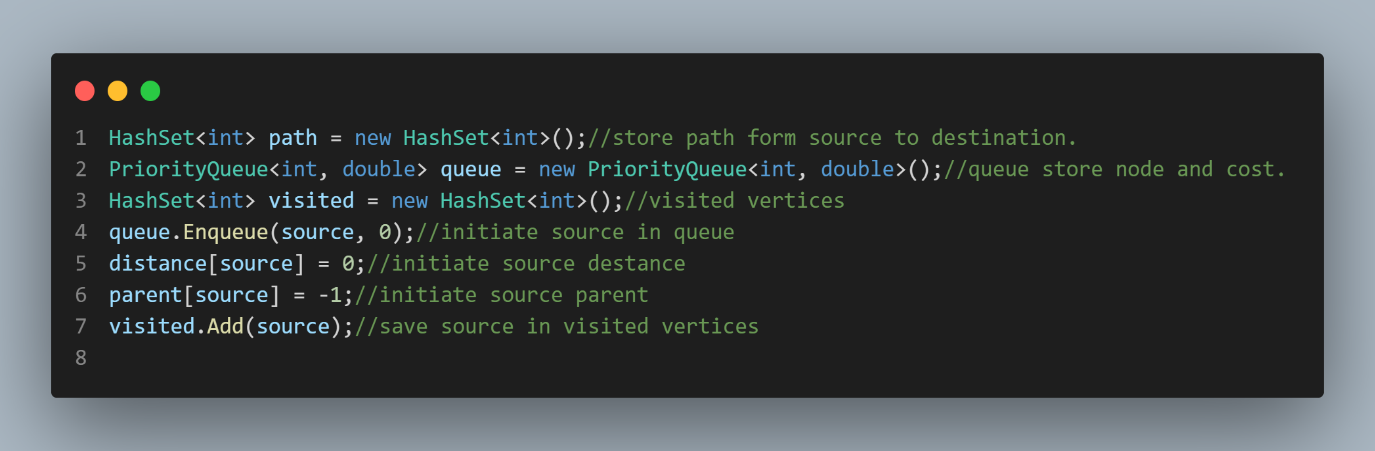
* **Initialize:**

Initialize O(1).

* **Initialize Parent :**

Initialize O(V).

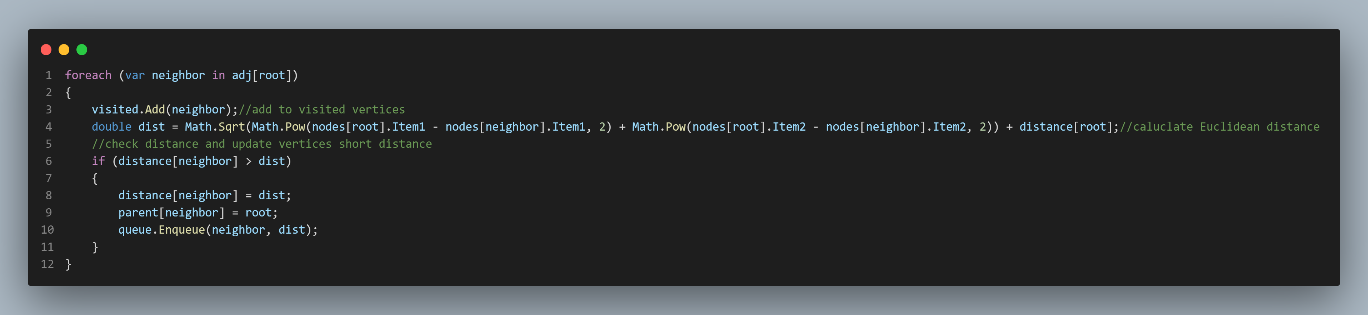
* **Initialize Priority Queue :**



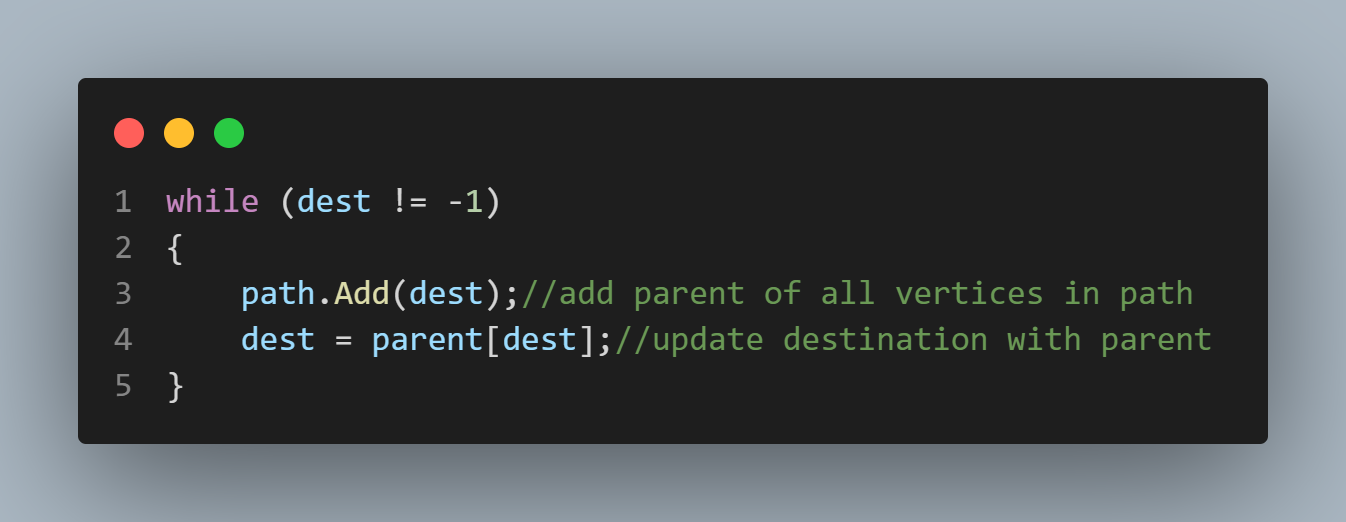
Insert in V times O(V)).

* **Loop All Vertices :**

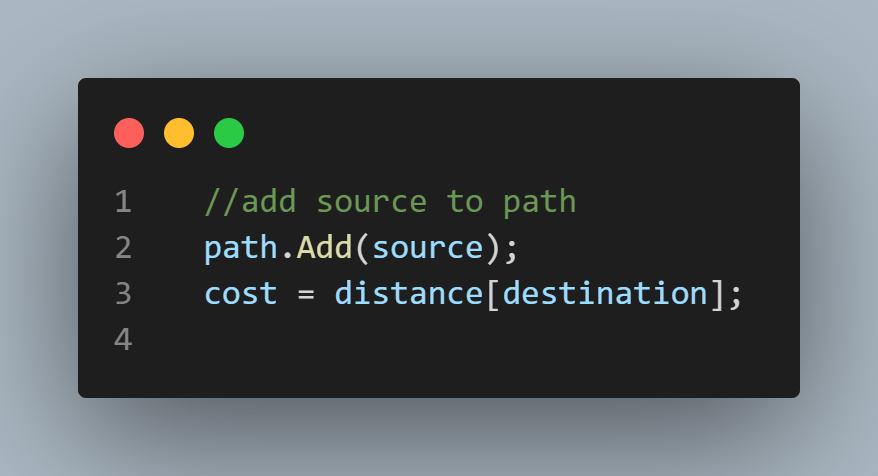
Loop On All Vertices O(V+E).

* **Loop On Adjacent:**

Loop on Adjacent Θ(E) E is Number Of Edges.

* **Loop Parent of Destination:**

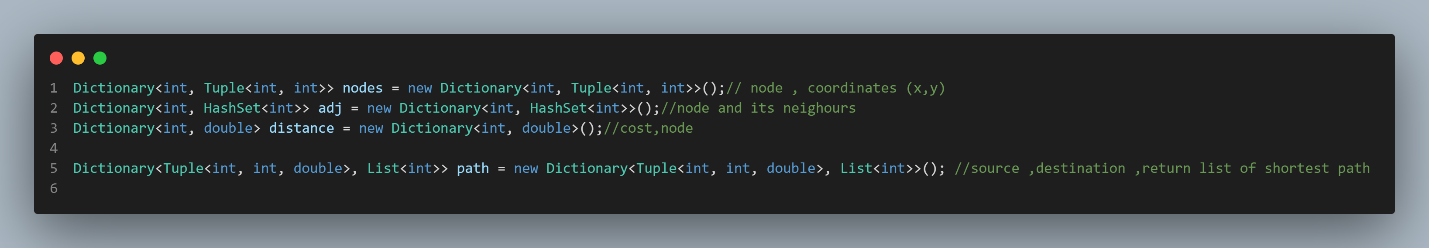
Loop On all Parent to Destination O(V).

* **Add Source to Path and Cost:**

Add Source to Path and Calculate Cost of Destination O(1).

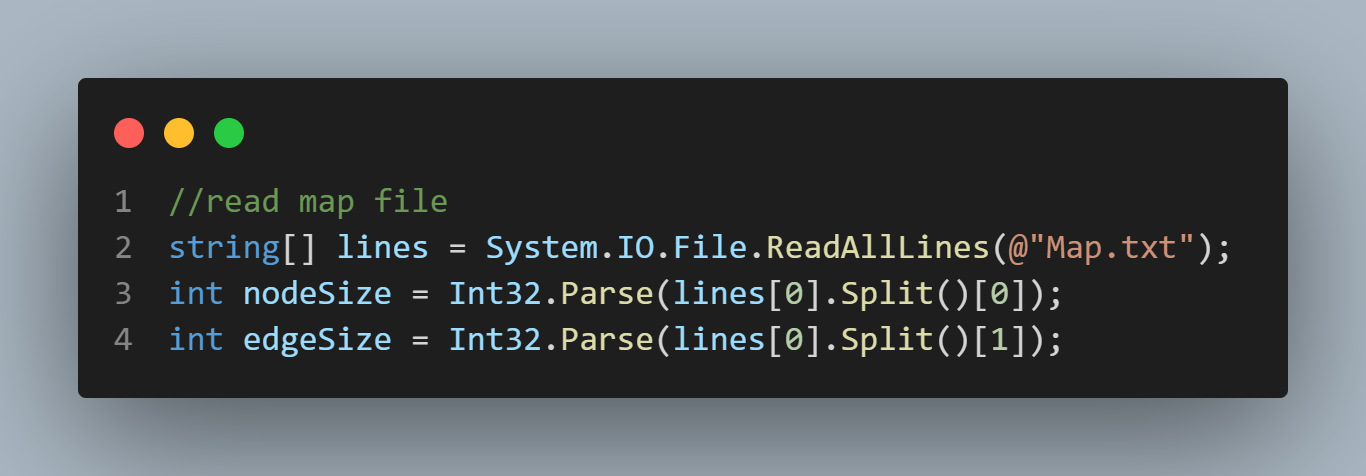
**Idea ( 1 ) :**

Stop Dijkstra when we Found Destination.

1. **Declaration Variables:**

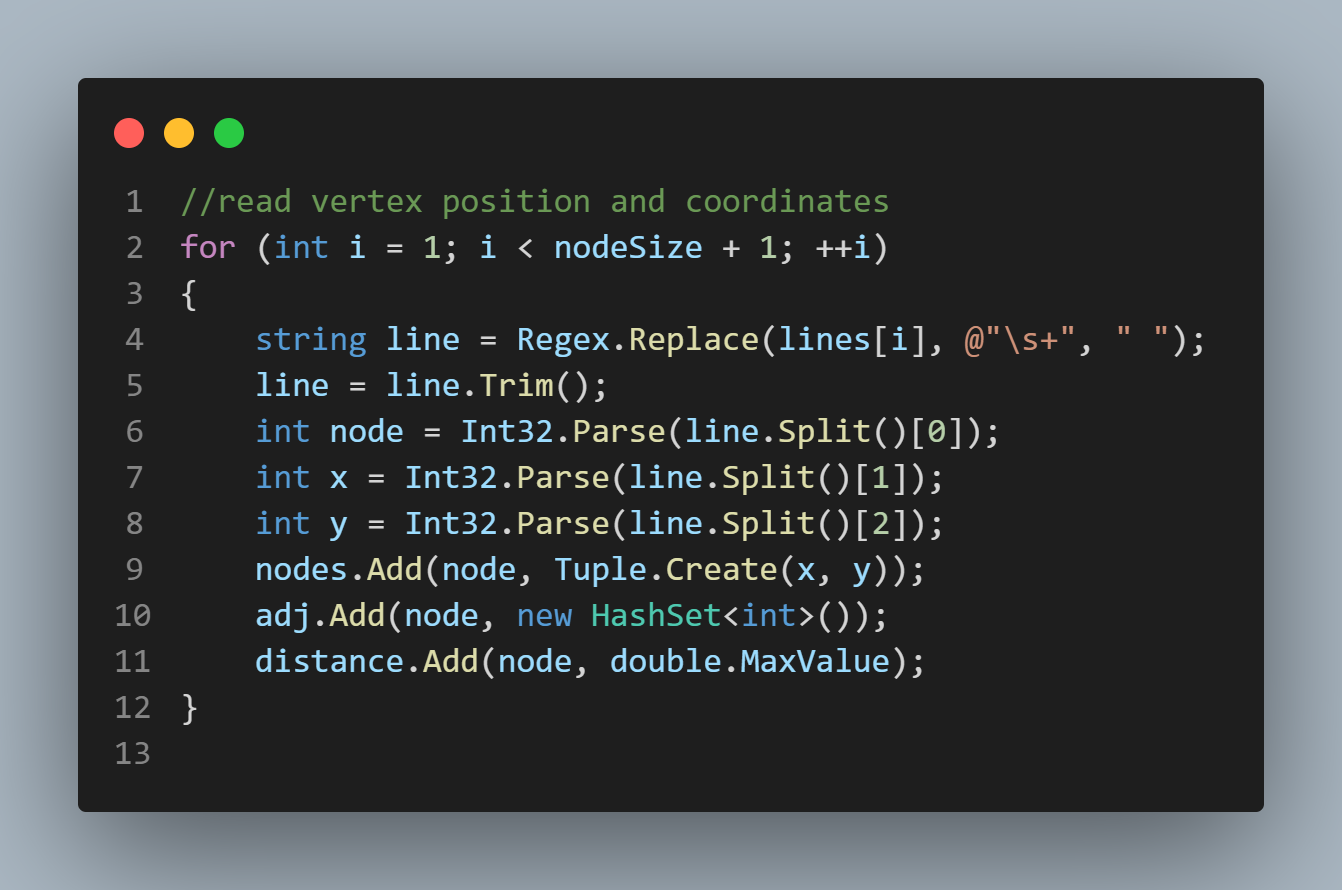
O( 1 ).

1. **Reading Map File :**

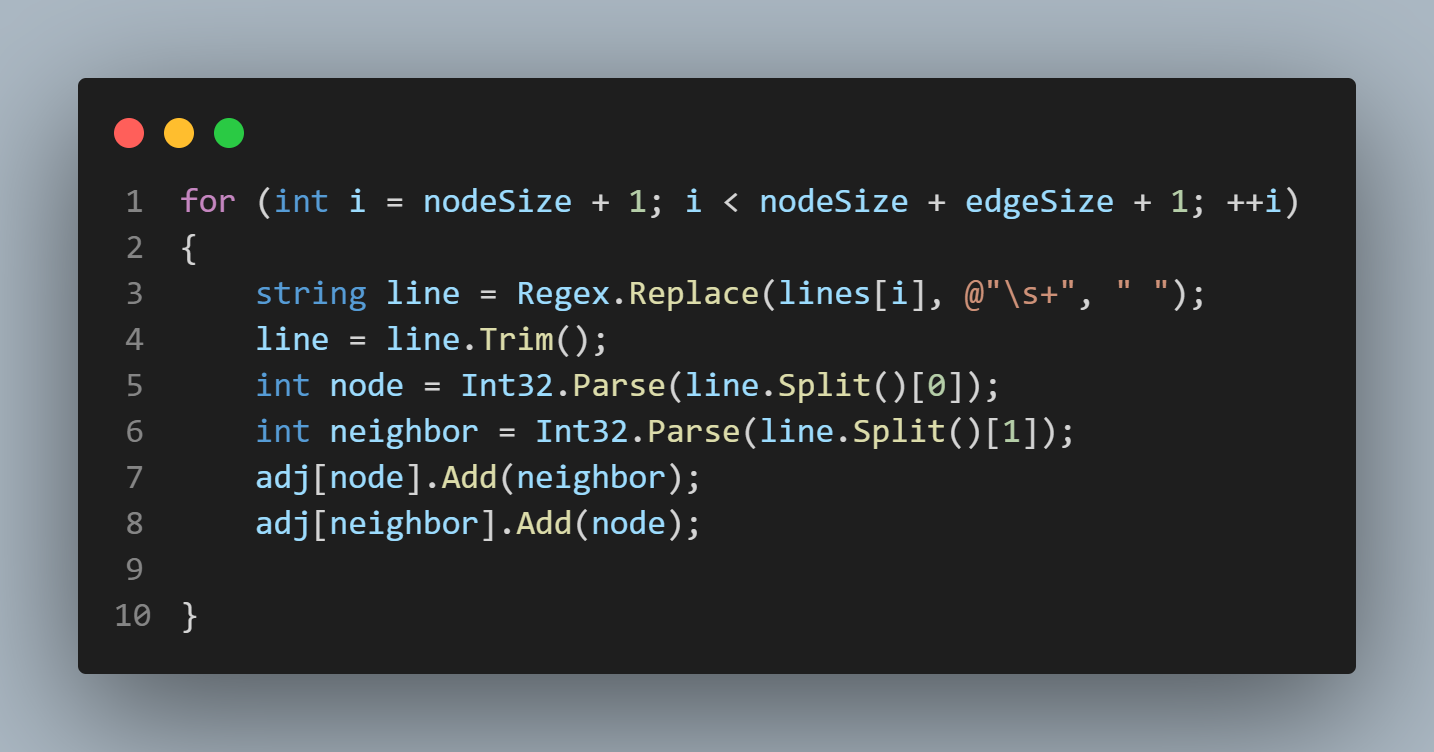


O( 1 ).

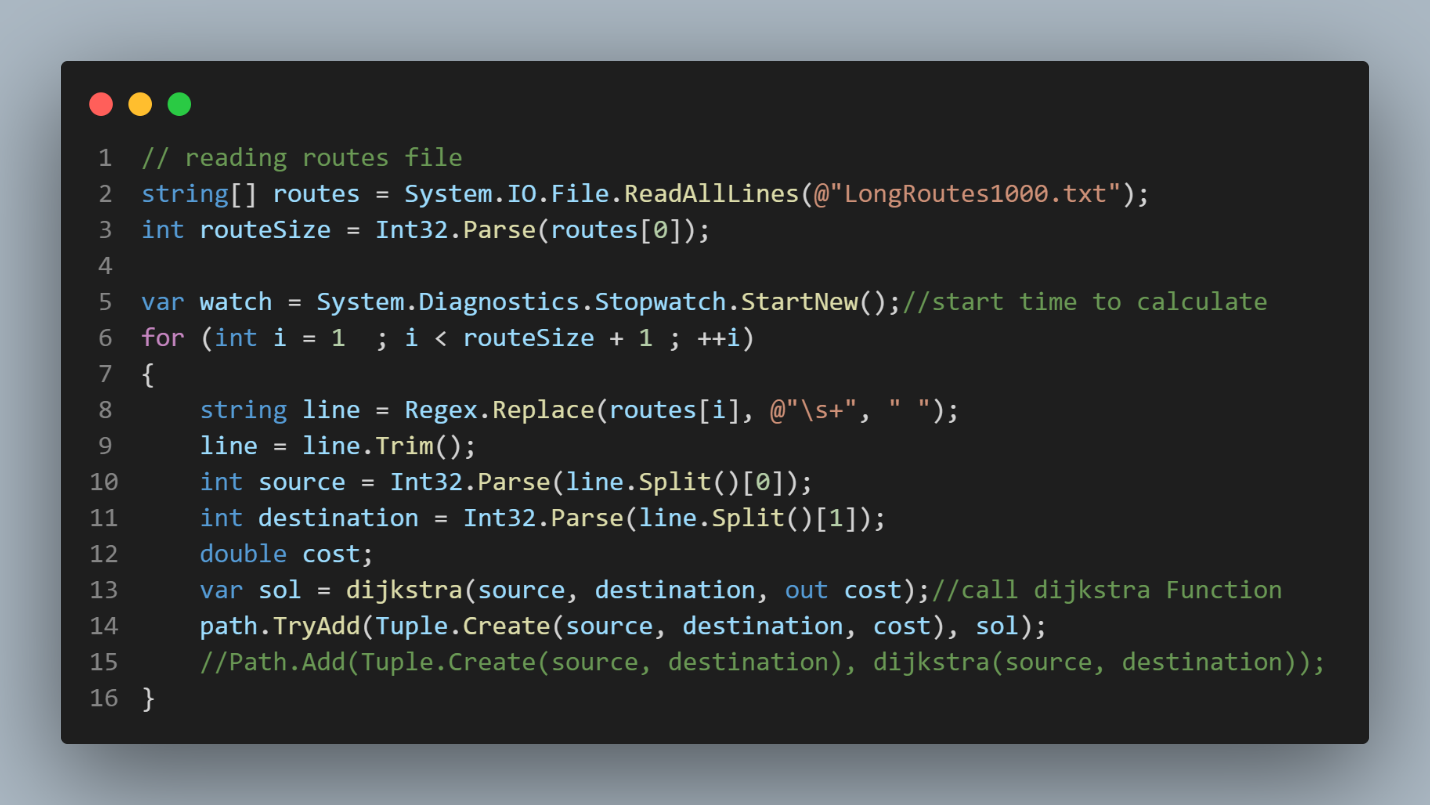
1. **Reading Vertices and Position:**



Θ(V ).

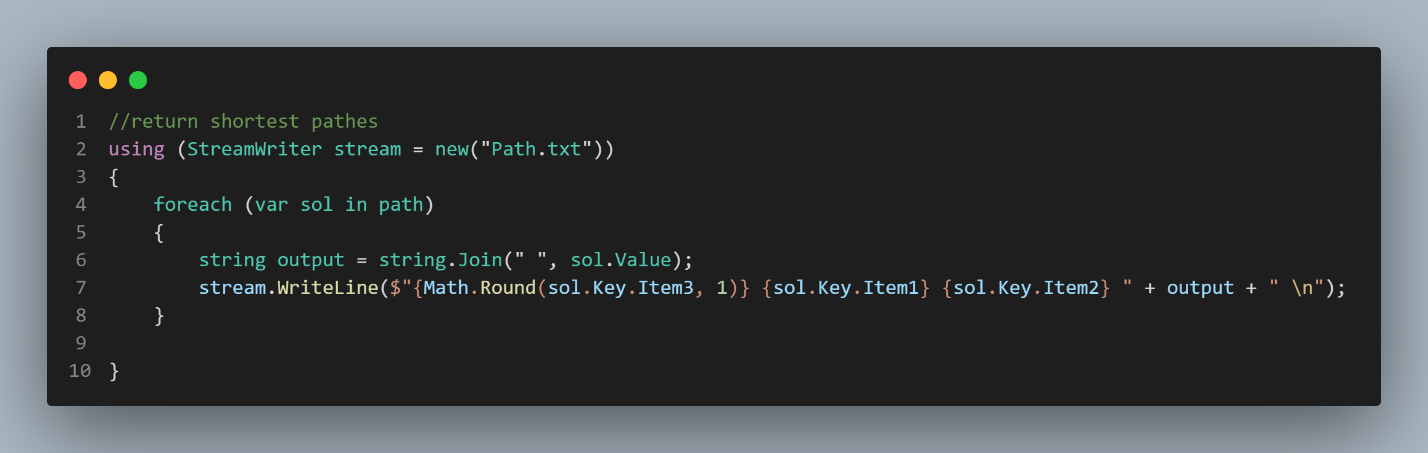
1. **Reading Edges :**

Θ( E )

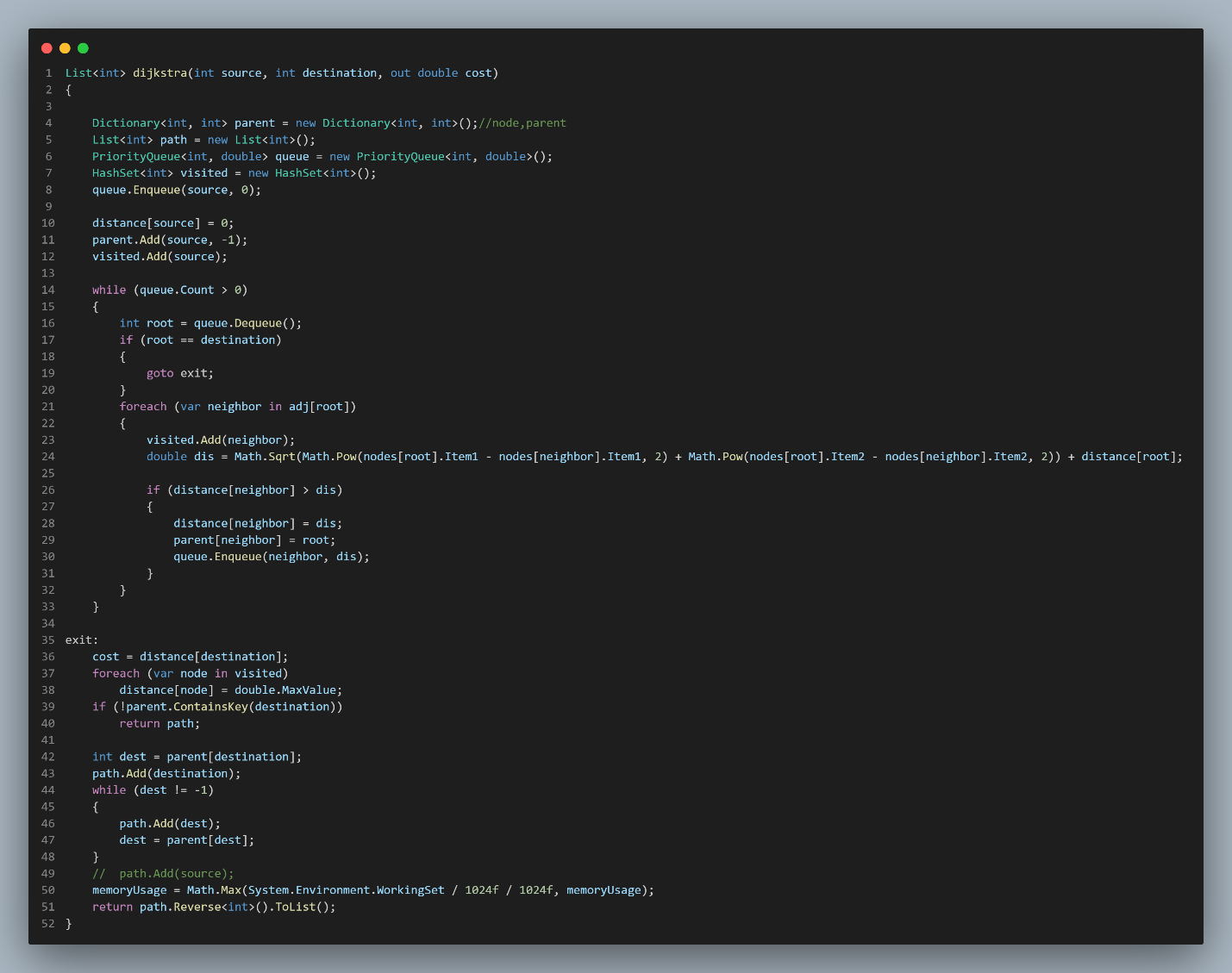
1. **Reading Routes File: **

O( R ).

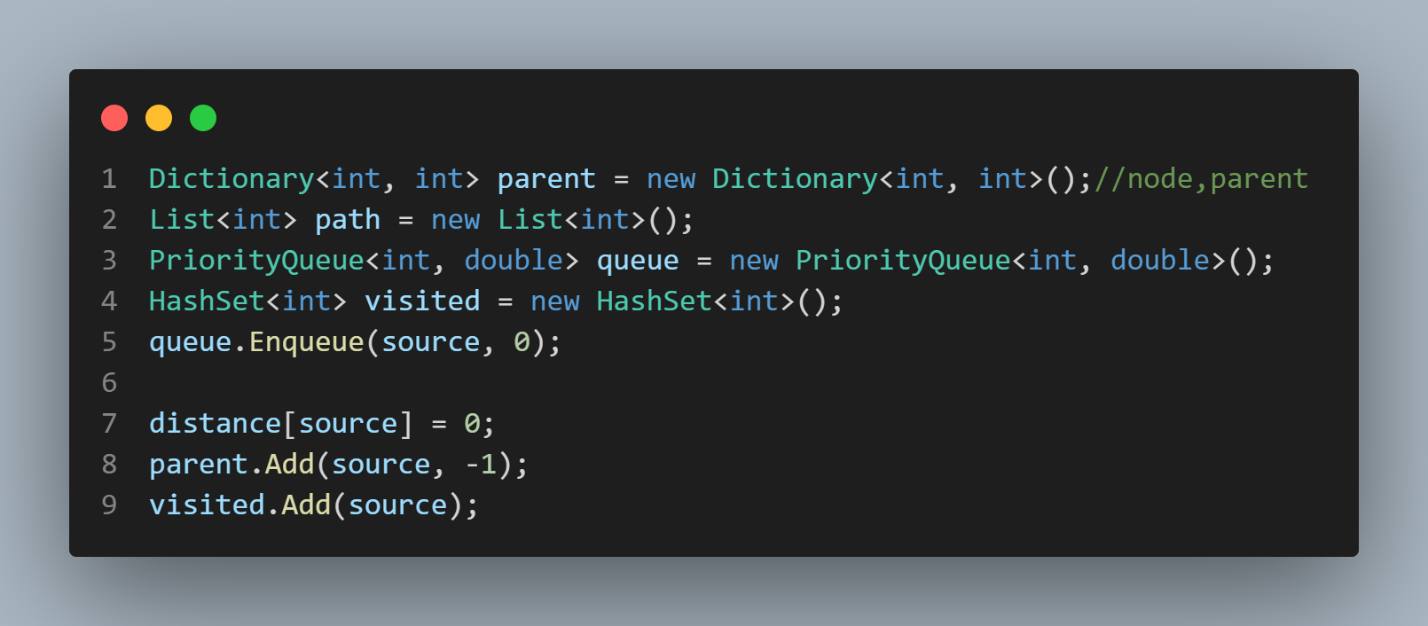
1. **Return Shortest Path :**

****O( R ).

1. **Dijkstra Function:**

****O( E ).

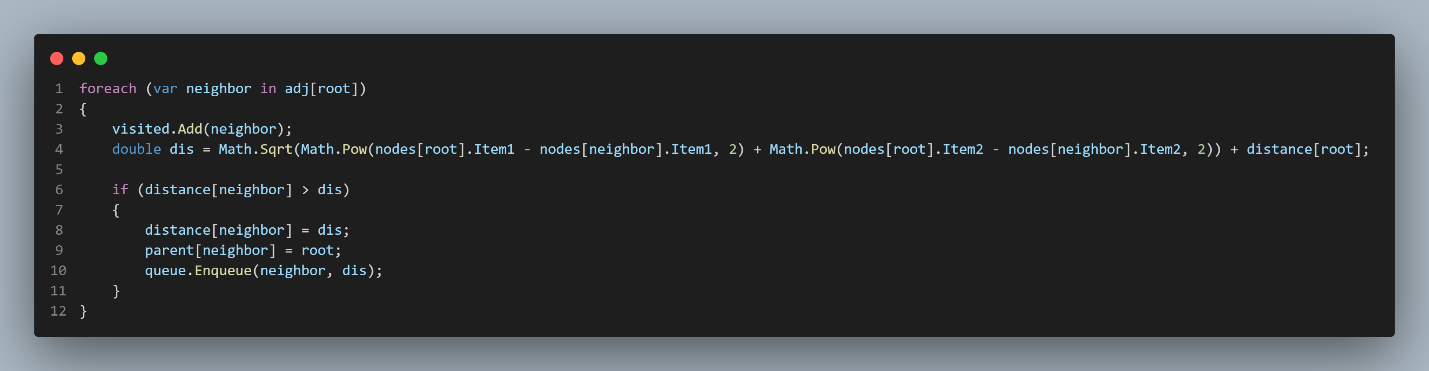
1. **Initialize:**

****O( 1 ).

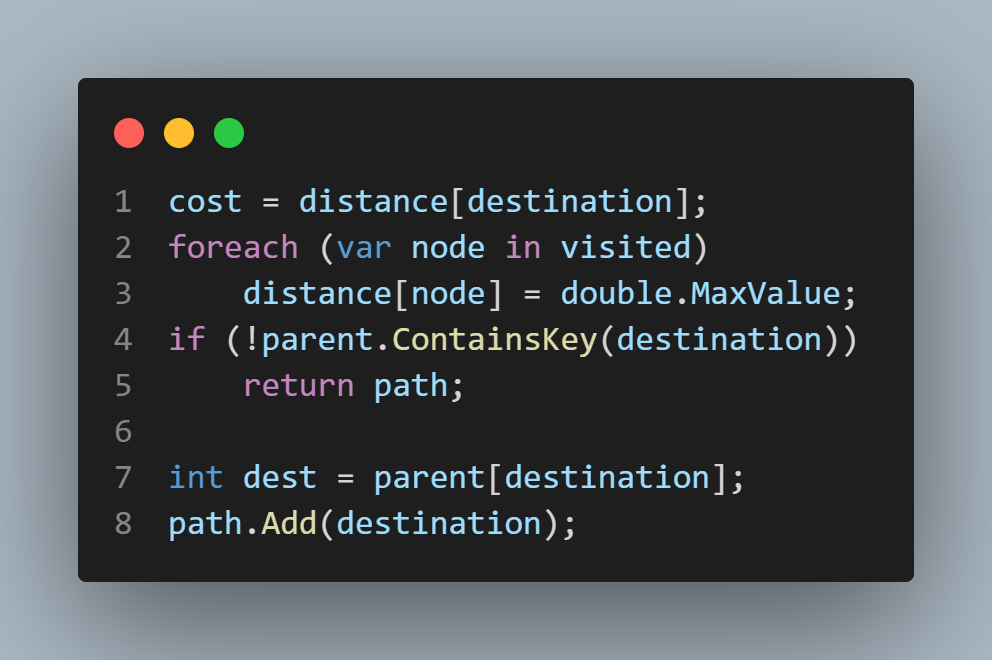
1. **Loop on Path:**

O( E ) .

1. **Loop On Neighbors:**

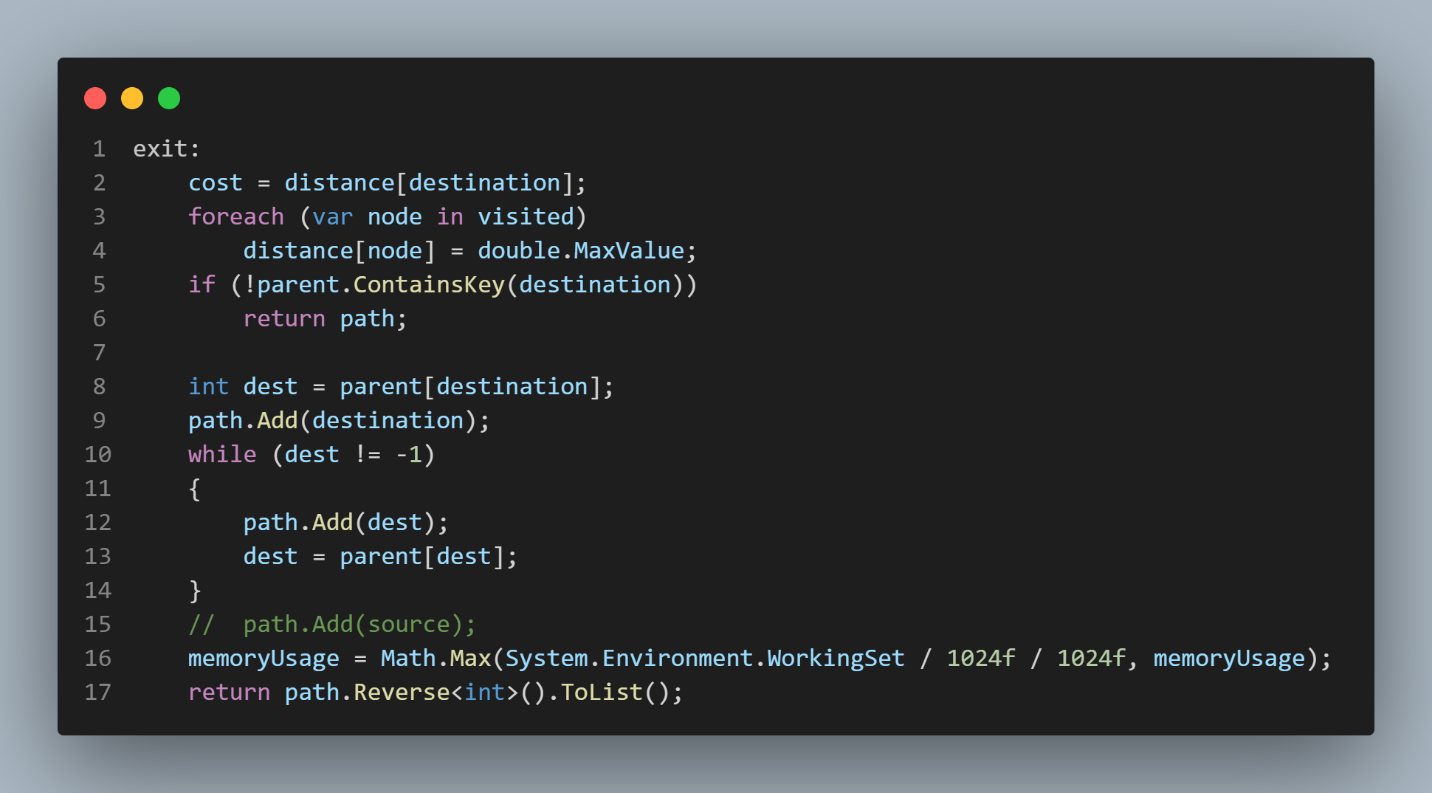
****O( V ).

1. **Loop on Visited Vertices:**

****O( V ).

1. **Loop On Vertices to Destination:**

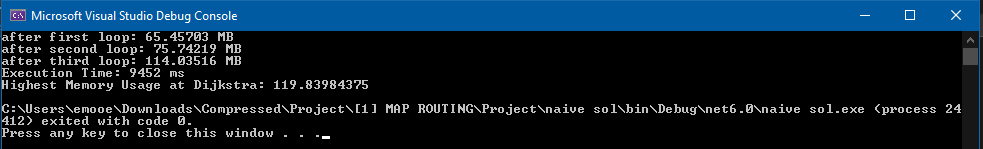
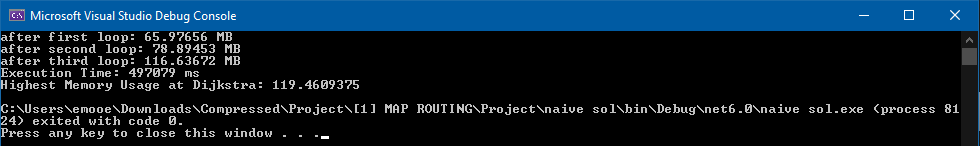
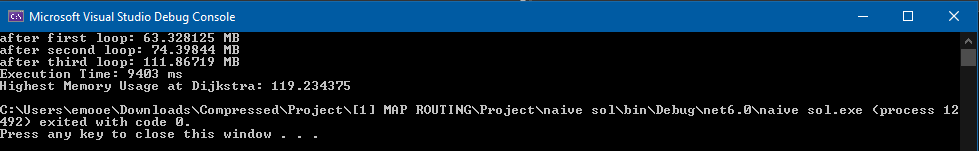
****O( V ).

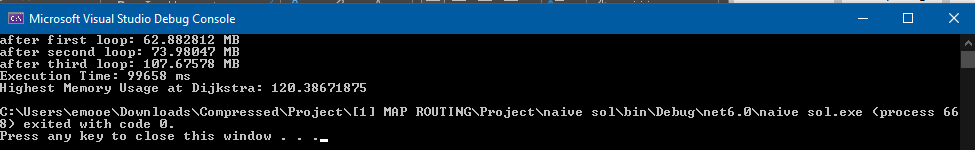
1. **Return Path:**

O( V ).

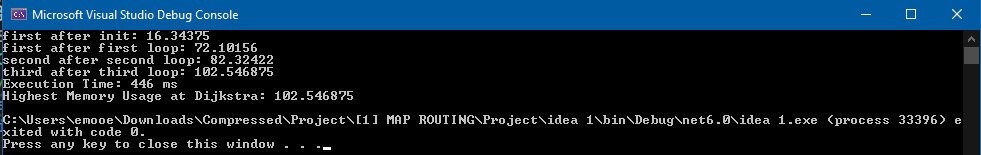
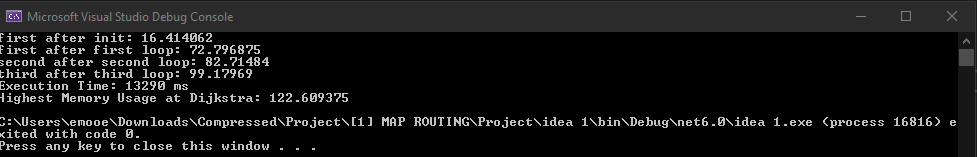
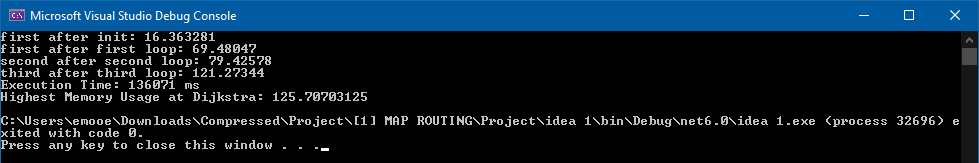
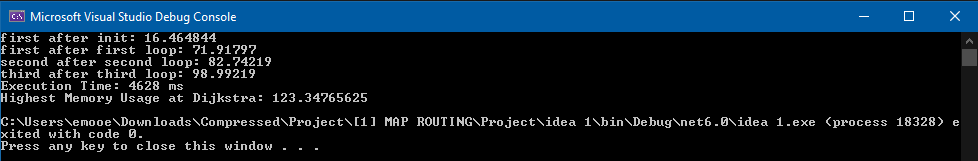
**Time Execution**

**Naive Execution:**

* **ShortRoutes100:**
* **ShortRoutes5000:**
* **LongRoutes100:**
* **LongRoutes1000:**

****

**Idea 1 Execution:**

* **ShortRoutes100:**
* **ShortRoutes5000:**
* **ShortRoutes50000:**
* **LongRoutes100:**
* **LongRoutes1000:**

****