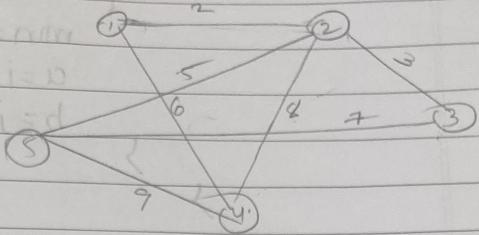


→ Kruskal's

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#define V 5
```



```
int parent[V];
```

```
int find(int i)
```

```
{
    while (parent[i] != i)
        i = parent[i];
    return i;
}
```

~~No update~~

```
> void unionb1(int i, int j)
```

```
{
    int a = find(i);
```

```
    int b = find(j);
```

```
    parent[a] = b;
```

```
> void kruskalMST(int cost[][V])
```

```
{
    int mincost = 0;
```

```
    for (int i=0; i<V; i++)
        parent[i] = i;
```

variables

```
& int edge_count = 0;
```

```
while (edge_count < V-1) {
```

```
    int min = INT_MAX, a = -1, b = -1;
```

```
    for (int i=0; i<V; i++) {
```

```
        for (int j=0; j<V; j++) {
```

```
if (find(i) != find(j) && cost[i][j] < min) {
    min = cost[i][j];
}
```

$a = i;$

$b = j;$

```
union1(a, b);
```

```
printf("Edge %d : (%d, %d) cost: %d \n", edge-count,
      a, b, min);
mincost += min;
```

```
printf("In Minimum cost=%d \n", mincost);
```

```
int main() {
```

```
int cost[10][10] = { {INT_MAX, 2, INT_MAX, 6, INT_MAX},  

                     {2, INT_MAX, 3, 8, 5},  

                     {INT_MAX, 3, INT_MAX, INT_MAX, 7},  

                     {6, 8, INT_MAX, INT_MAX, 9},  

                     {INT_MAX, 5, 7, 9, INT_MAX} };
```

```
KruskalMST(cost);
```

```
return 0;
```

```
Output:
```

edge 0 : (0,1) cost: 2

edge 1 : (1,2) cost: 3

edge 2 : (1,4) cost: 5

edge 3 : (0,3) cost: 6

11/07/24

classmate

Date _____
Page _____

→ Dijkstra's algorithm.

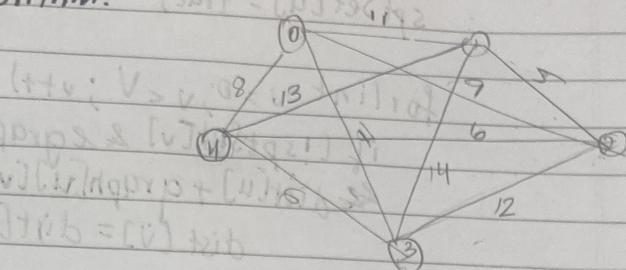
ALGORITHM

```
#include <limits.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define V 5
```



```
int minDistance(int dist[], bool sptSet[]) {
    int min = INT_MAX, min_index;
    for (int v = 0; v < V; v++)
        if (sptSet[v] == false && dist[v] <= min)
            min = dist[v], min_index = v;
    return min_index;
}
```

```
void printSolution(int dist[], int n)
```

```

    printf("vertex Distance from source\n");
    for (int i = 0; i < V; i++)
        printf("%d %d\n", i, dist[i]);
}
```

```
void dijkstra (int graph[V][V], int src)
```

N
W
H
M

```
int dist[V];
```

```
bool sptSet [V];
```

```
for (int i = 0; i < V; i++)
    dist[i] = INT_MAX, sptSet[i] = false;
```

```
dist[src] = 0;
```

```
for (int count = 0; count < V - 1; count++)
```

```
    int u = minDistance(dist, sptSet);
```

out put:

✓ vertere

०

2

3

M

Distance from source

8

1)

1

1

1

(V)nitib

7 + 82 + 82 = 100

9. *What is the name of the author?*

$(6 \div i + ni) \log$

$$z = \lim_{n \rightarrow \infty} z_n$$

-0.5

~~Section 1~~ + 6

100% H_2O_2

10. tri