


```

        min = key[v], min_index = v;

    return min_index;

}

void printMST(int parent[], int graph[V][V])

{

    cout<<"Edge \tWeight\n";

    for (int i = 1; i < V; i++)

        cout<<parent[i]<<" - "<<i<<" \t"<<graph[i][parent[i]]<<" \n";

}

void primMST(int graph[V][V])

{

    int parent[V];

    int key[V];

    bool mstSet[V];

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

        key[i] = INT_MAX, mstSet[i] = false;

    key[0] = 0;

```

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

{
    int u = minKey(key, mstSet);

    mstSet[u] = true;
    for (int v = 0; v < V; v++)

        if (graph[u][v] && mstSet[v] == false && graph[u][v] < key[v])

            parent[v] = u, key[v] = graph[u][v];

}
printMST(parent, graph);

}
```

#H572

۹۷۴۵۸۱

نمی‌تواند

④ انتفا ب عربیون نه این است که در قیاسیه فعلی است و عربی را انتفا ب گفتیم که زبان پادشاه آن در اول
در میان فعلی است که با قیاسیه است زبان شروع فعلی است و در قیاسیه زبان است و قیاسیه قبل
هست. ما باید فعلی است بر اساس زبان آن حاضر است و قیاسیه قیاسیه است و قیاسیه زبان
پادشاه پادشاه: ① قیاسیه است و در قیاسیه زبان آن حاضر است و قیاسیه قیاسیه
از آنکه مرتب در قیاسیه قبل انتفا ب است و در قیاسیه ۳ به قیاسیه آرایه است و مرتب که در قیاسیه
زبان را انتفا ب است و از زبان شروع فعلی است و در قیاسیه زبان است و قیاسیه قیاسیه
فعلی است و انتفا ب است و قیاسیه است و در قیاسیه زبان است و قیاسیه قیاسیه
زبان پادشاه زبان مرتب در قیاسیه

Q4

struct Activitiy

{

int start, finish;

};

bool activityCompare(Activitiy s1, Activitiy s2)

{

return (s1.finish < s2.finish);

```

}

void printMaxActivities(Activity arr[], int n)

{
    sort(arr, arr+n, activityCompare);

    cout << "Following activities are selected n";

    int i = 0;

    cout << "(" << arr[i].start << ", " << arr[i].finish << ")", ";
    for (int j = 1; j < n; j++)

    {

        if (arr[j].start >= arr[i].finish)

        {

            cout << "(" << arr[j].start << ", "

                << arr[j].finish << ")", ";

            i = j;

        }

    }
}

```

```
}
```

```
void SelectActivities(vector<int>s,vector<int>f){  
    vector<pair<int,int>>ans;  
  
    priority_queue<pair<int,int>,vector<pair<int,int>>,greater<pair<int,int>>>p;  
    for(int i=0;i<s.size();i++){  
  
        p.push(make_pair(f[i],s[i]));  
  
    }  
}
```

```
    auto it = p.top();
```

```
    int start = it.second;
```

```
    int end = it.first;
```

```
    p.pop();
```

```
    ans.push_back(make_pair(start,end));
```

```
    while(!p.empty()){
```

```
        auto itr = p.top();
```

```
p.pop();
```

```
if(itr.second >= end){
```

```
    start = itr.second;
```

```
    end = itr.first;
```

```
    ans.push_back(make_pair(start,end));
```

```
}
```

```
}
```

```
cout << "Following Activities should be selected. " << endl << endl;
```

```
for(auto itr=ans.begin();itr!=ans.end();itr++){
```

```
    cout << "Activity started at: " << (*itr).first << " and ends at " << (*itr).second << endl;
```

```
}
```

```
}
```

Q3

```
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;

}

int printSolution(int dist[], int n)

{

    printf("Vertex   Distance from Source\n");

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

        printf("%d\t\t %d\n", i, dist[i]);

}

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

{

    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);
```

```
    sptSet[u] = true;
```

```
    for (int v = 0; v < V; v++)
```

```
        if (!sptSet[v] && graph[u][v] && dist[u] != INT_MAX
```

```
            && dist[u] + graph[u][v] < dist[v])
```

```
            dist[v] = dist[u] + graph[u][v];
```

```
}
```

```
printSolution(dist, V);
```

```
}
```