main.c

Run

```c
#192210211
#include <stdio.h>
#include <stdlib.h>
#define inf 999999
int i,j,k,a,b,u,v,n,ne=1;
int min,mincost=0,p[9];
int cost[9][9] = {
    {0, 10, 20},
    {12, 0, 15},
    {16, 18, 0}
};
int applyfind(int);
int applyunion(int,int);
int applyfind(int i){
    while(p[i])
        i=p[i];
    return i;
}
int applyunion(int i,int j){
    if(i!=j) {
        p[j]=i;
        return 1;
    }
    return 0;
}
int main(){
    n = 3;
    printf("Minimum Cost Spanning Tree: \n");
    while(ne < n) {
        min = inf;
```

Output

Clear

```
/tmp/dKzoRx1qjh.o
Minimum Cost Spanning Tree:
1 -> 3
2 -> 3
```

```c
    p[j]=i;
    return 1;
  }
  return 0;
}
int main(){
  n = 3;
  printf("Minimum Cost Spanning Tree: \n");
  while(ne < n) {
    min = inf;
    for(i=1; i<=n; i++) {
      for(j=1; j <= n; j++) {
        if(cost[i][j] < min) {
          min=cost[i][j];
          a=u=i;
          b=v=j;
        }
      }
    }
    u=applyfind(u);
    v=applyfind(v);
    if(applyunion(u,v)) {
      printf("%d -> %d\n",a,b);
      mincost +=min;
    }
    cost[a][b]=cost[b][a]=999;
  }
  printf("\n\tMinimum cost = %d\n",mincost);
  return 0;
}
```

```
/tmp/dKzoRx1qjh.o
Minimum Cost Spanning Tree:
1 -> 3
2 -> 3
```