//// MST+BFS+DFS////

```
#include<bits/stdc++.h>
using namespace std;
char str[52][52];
int vis[52][52],number;
int dis[52][52];
#define mx 2500
int  par[mx+5];
int n;
int fx[]= {+1,-1,+0,+0};
int fy[]= {+0,+0,+1,-1};
int chk;
void dfs(int x, int y)
{
    if(x<0 || x>=n ||y<0 || y>=n ||str[x][y] == '.'|| vis[x][y]!=0)
        return;
    vis[x][y] = chk;
    for(int i=0; i<4; i++)
    {
        int xx = x + fx[i];
        int yy = y + fy[i];
        dfs(xx,yy);
    }
    return;
}

struct edge
{
    int  u,v,w;
    edge(int  a,int  b,int  c)
    {
        u = a,v = b,w = c;
    }
    bool operator <(const edge&p) const
    {
        return w<p.w;
vector<edge>mst1;
void bfs(int sx, int sy)
{
    queue<pair<int, int > >Q;
    Q.push(make_pair(sx,sy));
    dis[sx][sy] = 0;
    while(!Q.empty())
    {

        int uu = Q.front().first;
        int vv = Q.front().second;
        Q.pop();
        for(int i=0; i<4; i++)
        {
            int ff = uu + fx[i];
            int ss = vv + fy[i];
            if(ff>=0 && ff<n && ss>=0 && ss<n && dis[ff][ss] == -1)
            {
                dis[ff][ss] = dis[uu][vv] + 1;
                Q.push(make_pair(ff,ss));
                if(vis[ff][ss]>0 && vis[ff][ss]!=number)
                {
                    edge kk(number,vis[ff][ss],dis[ff][ss]);
                    mst1.push_back(kk);
                }
            }
        }
    }
}

int  findd(int  val)
{
    if(par[val] == val) return val;
    return par[val]=findd(par[val]);
}
void Union(int a,int b)
{
    par[findd(b)] = findd(a);
}
int mst(int  sz)
{
    int  s=0,nod = 0;
    sort(mst1.begin(),mst1.end());
    for(int i=0; i<=sz; i++)
        par[i] =i;
    for(int  i=0; i<(int )mst1.size(); i++)
    {
        int u = findd(mst1[i].u);
        int v =findd(mst1[i].v);
        if(u!=v)
        {
            Union(u,v);
            nod++;
            s+=mst1[i].w;
            if(nod == sz-1)
                break;
        }
    }
    return s;
}
int main()
{
    int t;
    scanf("%d",&t);
    while(t--){
    memset(vis,0,sizeof(vis));
    scanf("%d",&n);
    for(int i=0; i<n; i++)
        scanf("%s",str[i]);
    chk = 0;
    for(int i=0; i<n; i++)
        for(int j=0; j<n; j++)
        {
            if(str[i][j] == '*' && vis[i][j] == 0)
            {
                chk++;
                dfs(i,j);
            }
        }

    for(number = 1; number <= chk; number++)
    {
        for(int i=0; i<n; i++)
            for( int j =0; j< n; j++)
                if(vis[i][j] == number)
                {
                    memset(dis,-1,sizeof(dis));
                    bfs(i,j);
                }
    }
        int ans = mst(chk);
        printf("%d\n",ans);
        mst1.clear();
    }
    return 0;
}
```

```
10
.........*
..***...**
......****
..........
..*.....**
.....**.*.
******..**
..........
*******...
**********
....****..



11
```