

# 题解

- 一个炸弹覆盖的区域最多扩展到离它最近的硬方块（即#），所以可以考虑提取出每一行和每一列中不含#的最长连续字段。
- 由于上面的结论，每一个字符所在的**行内最长连续子段**和**列内最长连续子段**肯定最多只会被算一次。
- 所以可以考虑先把每一个最长连续子段缩成一个点，然后对原图中每个字符 $s[i][j]$ ，从它所在的**行内最长连续子段**向**列内最长连续子段**连一条边表示关系；
- 是不是有点熟悉？没错，最后做一遍二分图匹配就是答案了。
- 感觉字数好像不够，再写一行凑字数。

## 代码

```
#include<iostream>
#include<cstdio>
#include<queue>
#include<cstring>
using namespace std;

int n,m,t1=0,t2,k=0,head[3600],hang[60][60],lie[60][60],d[3600];
bool vis[3600];
char c,s[60][60];
struct edge{
    int u,v,w,next;
}e[100000];

void add(int u,int v,int w){
    e[k]=(edge){u,v,w,head[u]};
    head[u]=k++;
}

bool bfs()
{
    memset(vis,0,sizeof(vis));
    memset(d,0,sizeof(d));
    queue<int>que;
    que.push(0);
    vis[0]=1;
    while(!que.empty())
    {
        int u=que.front();que.pop();
        for(int i=head[u];~i;i=e[i].next)
        {
```

```

        int v=e[i].v,f=e[i].w;
        if(vis[v]||!f)continue;
        d[v]=d[u]+1;vis[v]=1;
        que.push(v);
    }
}
return d[t2+1];
}
int dfs(int u,int a)
{
    if(u==t2+1||a==0)return a;
    int ret=0,f;
    for(int i=head[u];~i;i=e[i].next)
    {
        if(d[u]+1!=d[e[i].v]||e[i].w<=0)continue;
        f=dfs(e[i].v,min(e[i].w,a-ret));
        if(f<=0)continue;
        e[i].w-=f;e[i^1].w+=f;
        ret+=f;
        if(!a)break;
    }
    return ret;
}
int dinic()
{
    int flow=0;
    while(bfs())
        flow+=dfs(0,0x7fffffff);
    return flow;
}

int main(){
    memset(head,-1,sizeof(head));
    scanf("%d%d",&n,&m);
    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++){
            cin>>c;
            s[i][j]=c;
        }
    for(int i=1;i<=n;i++){
        for(int j=1;j<=m;j++){
            if(s[i][j]!='#'){
                if(j==1||s[i][j-1]=='#')t1++;
                hang[i][j]=t1;
            }
        }
    }
    for(int i=1;i<=t1;i++)add(0,i,1),add(i,0,0);
    t2=t1;
    for(int j=1;j<=m;j++){
        for(int i=1;i<=n;i++)
            if(s[i][j]!='#'){
                if(i==1||s[i-1][j]=='#')t2++;

                lie[i][j]=t2;
            }
    }
}

```

```
    }  
}  
for(int i=t1+1;i<=t2;i++)add(i,t2+1,1),add(t2+1,i,0);  
for(int i=1;i<=n;i++)  
    for(int j=1;j<=m;j++)  
        if(s[i][j]!='#'&& s[i][j]!='x')  
            add(hang[i][j],lie[i][j],1),add(lie[i][j],hang[i][j],0);  
printf("%d",dinic());  
return 0;  
}
```