```cpp
/// https://toph.co/p/mr-savior
#include<bits/stdc++.h>
using namespace std;
char s[105][105];
int fx[] = {-1,+0,+0,+1};
int fy[] = {+0,-1,+1,+0};
int n,m,cost[105][105], vis[105][105];
struct dt{
    int x,y,w;
    bool operator < (const dt&p)const{
        return p.w<w;
    }
};
int BFS(){
    for(int i=0; i<101; i++){
        for(int j=0; j<101; j++){
            cost[i][j] = 10000000; vis[i][j] = 0;
        }
    }
    int sx,sy;
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            if(s[i][j]=='$'){
                sx = i, sy = j;
                break;
            }
        }
    }
    priority_queue<dt>qq;
    dt p; p.x = sx, p.y = sy, p.w = 0;
    qq.push(p);
    cost[sx][sy] = 0;

    while(!qq.empty()){
        dt u; u = qq.top();
        qq.pop();
        int x,y,w; x = u.x, y = u.y, w = u.w;
        if(vis[x][y]==1)continue;

        for(int k=0; k<4; k++){
            int tx = x+fx[k];
            int ty = y+fy[k];
            if(tx<0 || tx>=n || ty<0 || ty>=m){
                return w;
            }else{
                if(s[tx][ty]!='*'){
                    int nv=w+1;
                    if(s[tx][ty]=='.')nv=w;
                    if(s[tx][ty]=='#')nv=w+1;
                    if(nv<cost[tx][ty]){
                        dt z; z.x = tx, z.y = ty, z.w = nv;
                        cost[tx][ty] = nv;
                        qq.push(z);
                    }
                }
            }
        }
        vis[x][y]=1;
    }
    return -1;
}
int main(){
    int t; scanf("%d",&t);
    for(int ks=1; ks<=t; ks++){
        scanf("%d %d",&n,&m);
        for(int i=0; i<n; i++) scanf("%s",s[i]);
        int ans = BFS();
        if(ans==-1)printf("Case %d: Impossible\n",ks);
        else printf("Case %d: %d\n",ks,ans);
    }
    return 0;
}
```