

搜索

区间最值

滑动窗口

ST表格

bfs

bfs双向

bfs优先队列优化

dfs

meet-in-middle

搜索

区间最值

滑动窗口

```
#include<bits/stdc++.h>
using namespace std;
struct node{
    int val;
    int pos;
};
int ans[10000000];
int ans2[10000000];
node a[10000000];
deque<node>q;
deque<node>qq;
int main()
{
    int n;
    cin>>n;
    int k;
    cin>>k;
    for(int i=1;i<=n;i++)
    {
        cin>>a[i].val;
        a[i].pos=i-1;
    }
    for(int i=1;i<=n;i++)
    {
        while(!q.empty()&&q.back().val>=a[i].val)
        {
            q.pop_back();
        }
        q.push_back(a[i]);
        while(!q.empty()&&q.front().pos<i-k)
```

```

        {
            q.pop_front();
        }
        if(i>=k)ans[i]=q.front().val;
    }
    ans[0]=0;
    for(int i=k;i<=n;i++)
    {
        cout<<ans[i]<<" ";
    }
    cout<<endl;
    ans2[0]=0;
    for(int i=1;i<=n;i++)
    {
        while(!qq.empty()&&qq.back().val<=a[i].val)
        {
            qq.pop_back();
        }
        qq.push_back(a[i]);
        while(!qq.empty()&&qq.front().pos<i-k)
        {
            qq.pop_front();
        }
        if(i>=k)ans2[i]=qq.front().val;
    }
    for(int i=k;i<=n;i++)
    {
        cout<<ans2[i]<<" ";
    }
    return 0;
}

```

ST表格

```

#include<bits/stdc++.h>
using namespace std;
int dp[1000000][30];
inline int read()
{
    char c=getchar();int x=0,f=1;
    while(c<'0' || c>'9'){if(c=='-')f=-1;c=getchar();}
    while(c>='0'&&c<='9'){x=x*10+c-'0';c=getchar();}
    return x*f;
}
int solve(int l,int r)
{
    int k=log2(r-l+1);
    return max(dp[l][k],dp[r+1-(1<<k)][k]);
}
int main()

```

```

{
    ios::sync_with_stdio (false);
    int n,m;
    n=read();
    m=read();
    for(int i=1;i<=n;i++)
    {
        dp[i][0]=read();
    }
    for(int j=1;j<=21;j++)
    {
        for(int i=1;i+(1<<j)-1<=n;i++)
        {
            dp[i][j]=max(dp[i][j-1],dp[i+(1<<(j-1))][j-1]);
        }
    }
    for(int i=1;i<=m;i++)
    {
        int l,r;
        l=read();
        r=read();
        printf("%d\n",solve(l,r));
    }
    return 0;
}

```

bfs

bfs双向

```

#include <bits/stdc++.h>
using namespace std;
#define ll long long
ll n,m;
inline int read()
{
    int x=0,f=1;
    char c=getchar();
    while(c<'0' || c>'9')
    {
        if(c=='-') f=-1;
        c=getchar();
    }
    while(c>='0' && c<='9')
    {
        x=(x<<3)+(x<<1)+(c^48);
        c=getchar();
    }
    return x*f;
}

```

```

void write(int x){
    if(x>9) write(x/10);
    putchar(x%10^48);
}
struct node
{
    ll x;
    ll y;
};
ll qq;ll sx,sy,ex,ey,nx,ny;
deque<node>q;
const ll N=1010;
ll room[N][N];
ll dis[N][N];
ll vx[8]={-1,-1,0,1,1,1,0,-1};ll vy[8]={0,1,1,1,0,-1,-1,-1};
void dfs()
{
    ll x,y;

    while(!q.empty())
    {
        q.pop_back();
    }
    q.push_back((node){sx,sy});dis[sx][sy]=0;
    if(ex==sx&&ey==sy)
    {
        return ;
    }
    while(!q.empty())
    {
        x=q.front().x;
        y=q.front().y;
        q.pop_front();
        if(x==ex&&y==ey)
        {
            return ;
        }
        for(ll i=0;i<=7;i++)
        {
            nx=x+vx[i];
            ny=y+vy[i];
            if(1<=nx&&nx<=n&&1<=ny&&ny<=m)
            {
                if(dis[nx][ny]<=dis[x][y]+(room[x][y]!=i))
                {
                    continue;
                }
                dis[nx][ny]=dis[x][y]+(room[x][y]!=i);
                if(i==room[x][y])
                {
                    q.push_front((node){nx,ny});
                }
            }
        }
    }
}

```

```

        else
        {
            q.push_back((node){nx,ny});
        }
    }
}
}
}
int main()
{
    n=read();m=read();
    string s;
    for(11 i=1;i<=n;i++)
    {
        cin>>s;
        for(11 j=1;j<=m;j++)
        {
            room[i][j]=s[j-1]-'0';
        }
    }

    cin>>qq;
    for(11 i=1;i<=qq;i++)
    {
        sx=read();sy=read();ex=read();ey=read();
        memset(dis,0x3f,sizeof(dis));
        dfs();
        write(dis[ex][ey]);cout<<"\n";
    }
}

```

bfs优先队列优化

```

#include <bits/stdc++.h>
const int N=310;
char room[N][N];
int vis[N][N];
int n,m,q;
int dx[4]={0,0,1,-1};
int dy[4]={1,-1,0,0};
struct node
{
    int x;
    int y;
    int t;
    bool operator<(const node &a)const
    {
        return a.t<t;
    }
}

```

```

};
std::vector<std::pair<int,int>>>v[N][N];
std::priority_queue<node>qq;
int sx,sy,ex,ey,nx,ny;
int ans;
int bfs()
{
    while(!qq.empty())
    {
        node temp=qq.top();
        qq.pop();
        if(temp.x<0||temp.x>n-1||temp.y<0||temp.y>m-1||room[temp.x]
[temp.y]=='#'||vis[temp.x][temp.y])continue;
        if(temp.x==ex&&temp.y==ey)
        {
            return temp.t;
        }
        vis[temp.x][temp.y]=1;
        for(int i=0;i<4;i++)
        {
            nx=temp.x+dx[i];
            ny=temp.y+dy[i];
            qq.push({nx,ny,temp.t+1});
        }
        for(auto u:v[temp.x][temp.y])
        {
            qq.push({u.first,u.second,temp.t+3});
        }
    }
    return -1;
}
int main()
{
    int k;
    std::cin>>k;
    while(k--)
    {
        while(!qq.empty())qq.pop();
        std::cin>>n>>m>>q;
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<m;j++)
            {
                vis[i][j]=0;
                v[i][j].clear();
            }
        }

        std::string s;
        for(int i=0;i<n;i++)
        {

```

```

        std::cin>>s;
        for(int j=0;j<m;j++)
        {
            room[i][j]=s[j];
            if(room[i][j]=='S')
            {
                sx=i;sy=j;
            }
            if(room[i][j]=='T')
            {
                ex=i;ey=j;
            }
        }
    }
    qq.push({sx,sy,0});
    int x1,x2,y1,y2;
    for(int i=1;i<=q;i++)
    {
        std::cin>>x1>>y1>>x2>>y2;
        v[x1][y1].push_back({x2,y2});
    }
    for(auto u:v[sx][sy])qq.push({u.first,u.second,3});
    std::cout<<bfs()<<std::endl;
}

return 0;
}

```

dfs

meet-in-middle

```

#include <bits/stdc++.h>
using namespace std;
#define ll long long
const ll inf=0x7fffffff;
const ll N=2e5+10;
ll a[N];map<ll, ll>mp;
void solve()
{
    return ;
}
int main()
{
    //ios::sync_with_stdio(false);
    ll n,m;

    cin>>n>>m;
    a[0]=1;
    for(ll i=1;i<n;i++)
    {

```

```

        a[i]=a[i-1]*2;
    }
    for(int i=1;i<=m;i++)
    {
        int u,v;
        cin>>u>>v;
        --u;--v;
        a[u]|=((long long)1 << v);
        a[v]|=((long long)1 << u);
    }
    for(int i=0;i<(1<<(n/2));++i)
    {
        int cnt=0;
        int t=0;
        for(int j=0;j<n/2;++j)
        {
            if((i>>j)&1)
            {
                t=t^a[j];
                cnt++;
            }
        }
        if(!mp.count(t))
            mp[t]=cnt;
        else
            mp[t]=min(mp[t],cnt);
    }
    int ans=inf;
    for(int i=0;i<(1<<(n-n/2));i++)
    {
        int cnt=0;
        int t=0;
        for(int j=0;j<(n-n/2);j++)
        {
            if(((i>>j)&1))
            {
                t=t^a[j+n/2];
                cnt++;
            }
        }
        if(mp.count(t^(((1<<n)-1))))
            ans=min(ans,cnt+mp[t^(((1<<n)-1))]);
    }
    cout<<ans;
    return 0;
}

```