BFS

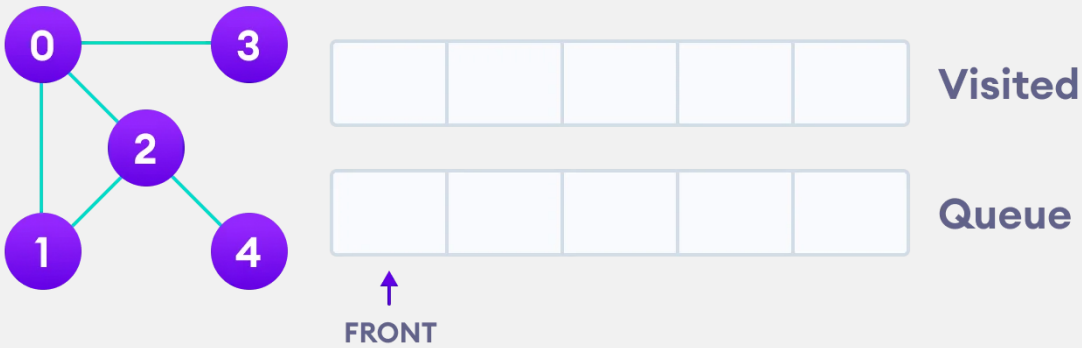
什么是BFS

> BFS全称为Breadth First Search，中文名广度优先搜索或宽度优先搜索。
>
> 广度优先即按层去遍历整个图，一层访问结束后再去访问下一层。
>
> 算法过程可以看做是图上火苗传播的过程：最开始只有起点着火了，在每一时刻，有火的节点都向它相邻的所有节点传播火苗。

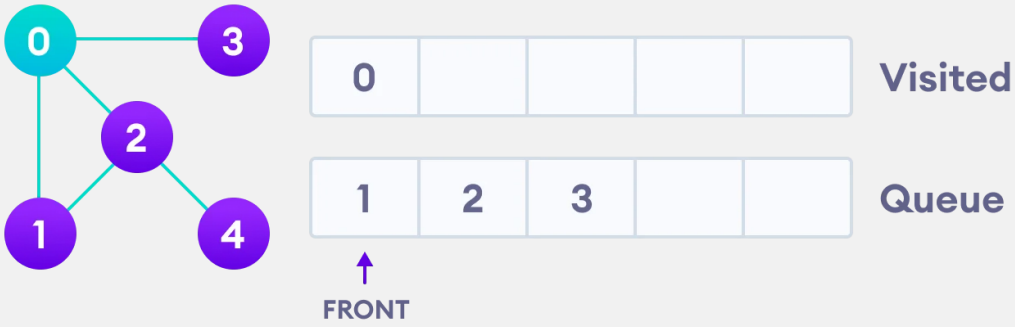
BFS算法流程

> 在标准的BFS中，所有的点被分为两种：
>
> > 1. 搜索过的
> > 2. 未搜索过的
>
> 这种处理避免了算法的死循环。
>
> 算法的流程如下：
>
> > 1. 从任意一个点开始搜索，把这个起始点放入队列
> > 2. 从队头取出一个点u，并把它标记为搜索过的
> > 3. 搜索点u的相邻点v,并把未被搜索过的点v加入队列
> > 4. 重复2，3步直至队列为空

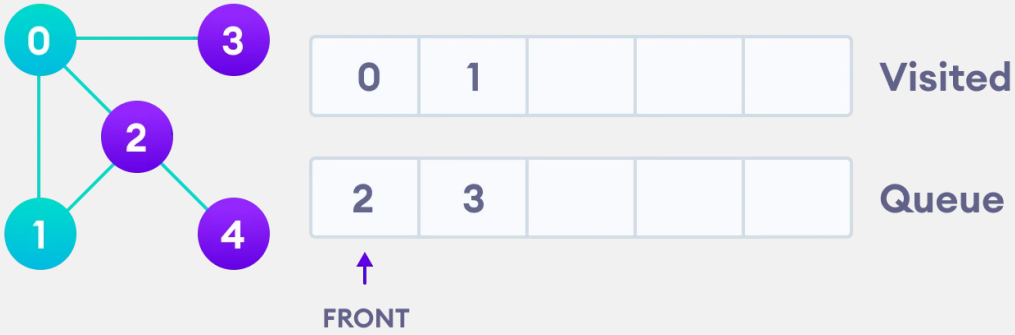
BFS算法流程举例

> 让我们看一个BFS案例，这个图上有5个点，从0号点开始BFS
>
> 
>

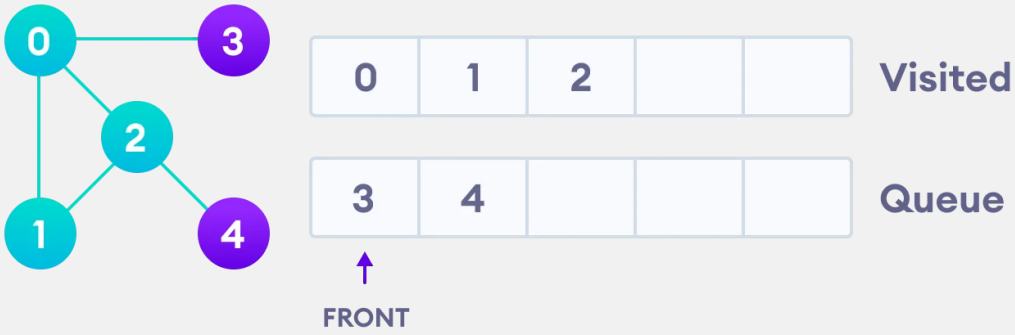
1. 取出队头，放入$Visited$列表，并把它相邻的点放入$Queue$



2. 接下来，我们从队列中取出1号点，将它放入$Visited$列表，并把相邻的未被访问(这里的访问包括在$Visited$列表和$Queue$中)的点放入队列。



3. 我们从队列中取出2号点，将它放入$Visited$列表，并把相邻的未被访问的点(4号点)放入队列。



4. 重复上述步骤直至队列为空

伪代码

```
bfs(s)
{
    q = new queue()
    q.push(s), visited[s] = true
    while (!q.empty())
    {
        u = q.pop()
        for each edge(u, v)
        {
            if (!visited[v])
            {
                q.push(v)
                visited[v] = true
            }
        }
    }
}
```

c++代码参考

```
#include<bits/stdc++.h>
using namespace std;
const int N = 110;
struct node{
    int x,y,step;
}que[N*N];
bool book[N][N];
int g[N][N],n,m;
int bfs()
{
    int head = 0,tail = -1;
    que[++tail] = {1,1,0};//这里选取(1,1)作为起始点
    int ne[4][2] = {1,0,0,1,0,-1,-1,0};
    while(tail>=head)//队列判空
    {
        auto temp = que[head++];//取出队头
        if(temp.x==n&&temp.y==m)//如果搜索到终点，就返回步数
            return temp.step;
        for(int i = 0 ; i < 4 ; i++)
        {
            int tx = temp.x+ne[i][0];//往四周走点
            int ty = temp.y+ne[i][1];
            if(book[tx][ty]||tx<1||ty<1||tx>n||ty>m||g[tx][ty]==1)continue;//判断
走到的点是否合法
            que[++tail] = {tx,ty,temp.step+1};//合法就加入到队列中
            book[tx][ty] = true;
        }
    }
    return -1;
}

int main()
{
    cin>>n>>m;
    for(int i = 1 ; i <= n ; i++)
    {
        for(int j = 1; j <= m ; j++)
        {
            cin>>g[i][j];
        }
    }
    cout<<bfs();
    return 0;
}
```

时空复杂度分析

> 在搜索时我们遍历了所有的边和点一遍，所以时间复杂度：$O(V+E)$ 易分析空间复杂度：$O(V)$

BFS练习题单

> - 洛谷
> - hdoj(个人博客中有BFS练习和题解)

skynesser的博客