针对aha算法的实现

因为学习到回溯法和分支限界法，所以特意回顾了这本书，实现了其中的相关代码

搜索

dfs的相关题目

遍历扑克

```
#include<bits/stdc++.h>

using namespace std;
int n;
int a[11];
bool book[11];
void dfs(int step){
    if(step==n+1){
        for(int i=1;i<=n;i++){
            cout<<a[i];
        }
        cout<<endl;
        return;
    }
    for(int i=1;i<=n;i++){
        if(book[i]==0){
            a[step]=i;
            book[i]=1;
            dfs(step+1);
            book[i]=0;
        }
    }
    return;
}
int main(){
    cin>>n;
    dfs(1);
    return 0;

}
```

遍历图

```
#include<bits/stdc++.h>

using namespace std;
#define N 51
vector<vector<int>> e(N,vector<int>(N,INT_MAX));
bool book[N];
int sum;
int n,m;
void dfs(int cur){
```

```
    cout<<cur<<endl;
    sum++;
    if(sum==n)return;
    for(int i=1;i<=n;i++){
        if(e[cur][i]==1&&book[i]==0){
            book[i]=1;
            dfs(i);
        }
    }
    return;
}
int main(){
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++){
            if(i==j)e[i][j]=0;

        }
    int a,b;
    for(int i=1;i<=m;i++){
        cin>>a>>b;
        e[a][b]=1;
        e[b][a]=1;
    }
    book[1]=1;
    dfs(1);
}
```

迷宫搜索

```
#include<iostream>
#include<climits>
using namespace std;
#define N 101
int a[N][N];
bool book[N][N];
int m,n;
int p,q;
int sp,sq;
int step;
int min1=INT_MAX;
int next1[4][2]={{0,1},{1,0},{0,-1},{-1,0}};
void dfs(int x,int y,int step){
    int nx,ny;
    if(x==p&&y==q){
        if(step<min1)min1=step;
        return;
    }
    for(int i=0;i<4;i++){

        nx=x+next1[i][0];
        ny=y+next1[i][1];
        if(nx<1||ny<1||nx>m||ny>n)continue;
        if(a[nx][ny]==0&&book[nx][ny]==0){
```

```
            book[nx][ny]=1;
            dfs(nx,ny,step+1);
            book[nx][ny]=0;
        }
    }
}

int main(){

    cin>>m>>n;
    for(int i=1;i<=m;i++)
        for(int j=1;j<=n;j++)cin>>a[i][j];
    cin>>sp>>sq>>p>>q;
    a[sp][sq]=1;
    dfs(sp,sq,0);
    cout<<min1<<endl;
}
```

水管工游戏

```
#include<bits/stdc++.h>

using namespace std;
#define N 51
int a[N][N];
bool book[N][N];
struct node{
    int x;
    int y;
}s[100];
int m,n;
int flag=0;
int top=0;
void dfs(int x,int y,int f){
    if(x==m&&y==n+1){
        flag=1;
        for(int i=1;i<=top;i++)cout<<"("<<s[i].x<<","<<s[i].y<<")";
        return;
    }
    if(x<1||y<1||x>m||y>n)return;
    if(book[x][y]==1)return;
    book[x][y]=1;
    top++;
    s[top].x=x;
    s[top].y=y;

    if(a[x][y]==5||a[x][y]==6){
        if(f==1)dfs(x,y+1,1);
        if(f==2)dfs(x+1,y,2);
        if(f==3)dfs(x,y-1,3);
        if(f==4)dfs(x-1,y,4);
    }else if(a[x][y]==0);
    else{
        if(f==1){
```

```
            dfs(x+1,y,2);
            dfs(x-1,y,4);
        }
        if(f==2){
            dfs(x,y-1,3);
            dfs(x,y+1,1);
        }
        if(f==3){
            dfs(x+1,y,2);
            dfs(x-1,y,4);
        }
        if(f==4){
            dfs(x,y-1,3);
            dfs(x,y+1,1);
        }
        }
        book[x][y]=0;
        top--;

}


int main(){
    cin>>m>>n;
    for(int i=1;i<=m;i++)
        for(int j=1;j<=n;j++)cin>>a[i][j];

    dfs(1,1,1);
    if(flag==0)cout<<"impossible"<<endl;
}
```

bfs的相关题目

走迷宫

```
#include<bits/stdc++.h>

using namespace std;
#define N 51
#define N2 50*50+1
struct node{
    int x;
    int y;
    int f;
    int s;
};

int main(){
    struct node que[N2];
    int a[N][N];
    bool book[N][N];
    int next1[4][2]={{0,1},{1,0},{0,-1},{-1,0}};
    int head,tail;
    int flag=0;//标记是否找到
```

```
    int m,n,sp,sq,p,q;
    int min1;
    cin>>m>>n;
    for(int i=1;i<=m;i++)
        for(int j=1;j<=n;j++)cin>>a[i][j];
    cin>>sp>>sq>>p>>q;
    head=1;
    tail=1;
    que[head].x=sp;
    que[head].y=sq;
    que[head].f=0;
    que[head].s=0;
    tail++;
    int nx,ny;
    while(head<tail&&!flag){
        for(int i=0;i<4;i++){
            nx=que[head].x+next1[i][0];
            ny=que[head].y+next1[i][1];
            if(nx<1||ny<1||nx>m||ny>n)continue;
            if(a[nx][ny]==0&&book[nx][ny]==0){
                que[tail].x=nx;
                que[tail].y=ny;
                que[tail].f=head;
                que[tail].s=que[head].s+1;
                tail++;
            }
            if(nx==p&&ny==q){
                flag=1;
                break;
            }
        }
        head++;
    }
    cout<<que[tail-1].s<<endl;
}
```

遍历图

```
#include<bits/stdc++.h>

using namespace std;
#define N 51
vector<vector<int>> e(N,vector<int>(N,INT_MAX));
bool book[N];
int m,n;
int main(){
    cin>>m>>n;
    for(int i=1;i<=m;i++)
        for(int j=1;j<=m;j++)if(i==j)e[i][j]=0;
    int a,b;
    for(int i=1;i<=n;i++){
        cin>>a>>b;
        e[a][b]=1;
        e[b][a]=1;
```

```
    }

    int que[10001];
    int head=1;
    int tail=1;
    que[head]=1;
    tail++;
    book[1]=1;
    int cur;
    while(head<tail){
        cur=que[head];
        for(int i=1;i<=m;i++){
            if(e[cur][i]==1&&!book[i]){
                que[tail]=i;
                book[i]=1;
                tail++;
            }
            if(tail>m)break;
        }
        head++;
    }

    for(int i=1;i<tail;i++)cout<<que[i];
}
```

最少转机

```
#include<iostream>
#include<vector>
#include<queue>
using namespace std;
#define N 101
vector<vector<int>> e(N,vector<int>(N,INT_MAX));
struct node{
    bool book;
    int front;
    int size;
}no[N];

int n,m;
queue<int> q;
int s,f;
int bfs(int goal){
    int fir=q.front();
    no[fir].size=0;
    while(!q.empty()){
        if(no[goal].book==1)return no[goal].size;
        int cur=q.front();
        q.pop();
        for(int i=1;i<=m;i++){
            if(e[cur][i]==1&&no[i].book==0){
                q.push(i);
                no[i].book=1;
                no[i].front=cur;
```

```
                no[i].size=no[cur].size+1;
            }
        }
    }
    return -1;
}
int main(){
    cin>>m>>n;
    cin>>s>>f;
    for(int i=1;i<=m;i++)
        for(int j=1;j<=m;j++)if(i==j)e[i][j]=0;
    int a,b;
    for(int i=1;i<=n;i++){
        cin>>a>>b;
        e[a][b]=1;
        e[b][a]=1;
    }
    q.push(s);
    no[s].book=1;
    int min1=bfs(f);
    if(min1==-1)cout<<"impossible"<<endl;
    else cout<<min1;
}
```

最短路径

Flord

```
#include<bits/stdc++.h>

using namespace std;
#define N 501
int n,m;
int t1,t2,t3;
vector<vector<int>> e(N,vector<int>(N,INT_MAX));
int main(){
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            if(i==j)e[i][j]=0;
    for(int i=1;i<=m;i++){
        cin>>t1>>t2>>t3;
        e[t1][t2]=t3;
    }
    for(int k=1;k<=n;k++)
        for(int i=1;i<=n;i++)
            for(int j=1;j<=n;j++)
                if(e[i][k]+e[k][j]<e[i][j]&&e[i][k]<INT_MAX&&e[k][j]<INT_MAX)e[i]
[j]=e[i][k]+e[k][j];
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
        cout<<e[i][j]<<" ";
```

```
    cout<<endl;
}
```

dijkstra

```
#include<bits/stdc++.h>

using namespace std;
#define N 51
vector<vector<int>> e(N,vector<int>(N,INT_MAX));
int n,m;
int t1,t2,t3;
bool book[N];
vector<int> dis(N);
int main(){
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            if(i==j)e[i][j]=0;
    for(int i=1;i<=m;i++){
        cin>>t1>>t2>>t3;
        e[t1][t2]=t3;
    }
    for(int i=1;i<=n;i++)book[i]=0;
    for(int i=1;i<=n;i++)dis[i]=e[1][i];
    book[1]=1;
    for(int i=1;i<=n-1;i++){
        int min1=INT_MAX;
        int u=0;
        for(int j=1;j<=n;j++){
            if(book[j]==0&&dis[j]<min1){
                min1=dis[j];
                u=j;
            }
        }
        for(int j=1;j<=n;j++){
            if(e[u][j]<INT_MAX&&dis[u]+e[u][j]<dis[j])dis[j]=dis[u]+e[u][j];
        }
        book[u]=1;
}
        for(int i=1;i<=n;i++)cout<<dis[i]<<" ";


}
```

Bellman-ford

```
#include<bits/stdc++.h>

using namespace std;
#define ll long long
```

```
const int N=51;
ll dis[N];
ll u[N],v[N],w[N];
bool check,flag;
int m,n;
int main(){
    cin>>n>>m;
    for(int i=1;i<=m;i++)cin>>u[i]>>v[i]>>w[i];
    for(int i=1;i<=n;i++)dis[i]=INT_MAX;
    dis[1]=0;
    for(int k=1;k<=n-1;k++){
        check=0;
        for(int i=1;i<=m;i++){
            if(dis[v[i]]>dis[u[i]]+w[i]){
                dis[v[i]]=dis[u[i]]+w[i];
                check=1;
            }
        }
        if(check==0)break;
    }
    flag=0;
    for(int i=1;i<=m;i++){
        if(dis[v[i]]>dis[u[i]]+w[i])flag=1;
    }
    if(flag==1)cout<<"有负路"<<endl;
    else{
        for(int i=1;i<=n;i++)cout<<dis[i]<<" ";
    }



}
```

队列形式Bellman-ford

```
#include<bits/stdc++.h>

using namespace std;
#define ll long long
const int N=51;
ll dis[N];
ll u[N],v[N],w[N];
ll first[N],next1[N];
bool check,flag;
bool book[N];
int m,n;
int main(){
    cin>>n>>m;
    for(int i=1;i<=n;i++)dis[i]=INT_MAX;
    dis[1]=0;
    for(int i=1;i<=n;i++)book[i]=0;
    for(int i=1;i<=n;i++)first[i]=-1;
    for(int i=1;i<=m;i++){
        cin>>u[i]>>v[i]>>w[i];
```

```
        next1[i]=first[u[i]];
        first[u[i]]=i;
    }
    queue<int> s;
    s.push(1);
    book[1]=1;
    int top;
    int no;
    while(!s.empty()){
        top=s.front();
        no=first[top];
        while(no!=-1){
            if(dis[v[no]]>dis[u[no]]+w[no]&&dis[u[no]]<INT_MAX){
                dis[v[no]]=dis[u[no]]+w[no];
                if(book[v[no]]==0){
                    s.push(v[no]);
                    book[v[no]]=1;
                }
            }
            no=next1[no];
        }
        s.pop();
        book[top]=0;
    }
    for(int i=1;i<=n;i++){
        cout<<dis[i]<<" ";
    }
}
```

最小生成树

kruskal

```
#include<bits/stdc++.h>

using namespace std;
#define N 101
struct edge{
    int u;
    int v;
    int w;
};
struct edge e[N];
int n,m;
int f[N];
int sum,count1=0;
bool cmp(const edge &a,const edge &b){
    return a.w<b.w;
}
int getf(int x){
    if(f[x]==x)return x;
    else{
```

```
        f[x]=getf(f[x]);
        return f[x];
    }
}
int merge(int x,int y){
    int t1=getf(x);
    int t2=getf(y);
    if(t1!=t2){
        f[t2]=t1;
        return 1;
    }
    return 0;
}
int main(){
    cin>>n>>m;
    for(int i=1;i<=m;i++)cin>>e[i].u>>e[i].v>>e[i].w;
    sort(e+1,e+m+1,cmp);
    for(int i=1;i<=n;i++)f[i]=i;
    for(int i=1;i<=m;i++){
        if(merge(e[i].u,e[i].v)){
            count1++;
            sum+=e[i].w;
        }
        if(count1==n-1)break;
    }
    cout<<sum<<endl;
}
```

prim

```
#include<bits/stdc++.h>

using namespace std;
#define N 101
int count1=0,sum=0;
int n,m;
int e[N][N];
int dis[N];
int book[N]={0};
int min1;
int j;
int main(){
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
        if(i==j)e[i][j]=0;
        else e[i][j]=INT_MAX;
    int t1,t2,t3;
    for(int i=1;i<=m;i++){
        cin>>t1>>t2>>t3;
        e[t1][t2]=t3;
        e[t2][t1]=t3;
    }
    for(int i=1;i<=n;i++)dis[i]=e[1][i];
```

```
    book[1]=1;
    count1++;
    while(count1<n){
        min1=INT_MAX;
        for(int i=1;i<=n;i++){
            if(book[i]==0&&dis[i]<min1){
                min1=dis[i];
                j=i;
            }
        }
        book[j]=1;
        count1++;
        sum+=dis[j];
        for(int i=1;i<=n;i++){
            if(book[i]==0&&dis[i]>e[j][i])dis[i]=e[j][i];
        }
    }
    cout<<sum<<endl;
}
```

并查集

```
#include<bits/stdc++.h>

using namespace std;
#define N 101
int f[N];
int n,m,sum;
int getf(int x){
    if(f[x]==x)return x;
    else {
        f[x]=getf(f[x]);
        return f[x];
    }
}
void merge(int x,int y){
    int t1,t2;
    t1=getf(x);
    t2=getf(y);
    if(t1!=t2){
        f[t2]=t1;
    }
}
int main(){
    cin>>n>>m;
    int x,y;
    for(int i=1;i<=n;i++)f[i]=i;

    for(int i=1;i<=m;i++){
        cin>>x>>y;
        merge(x,y);
```

```
    }
    for(int i=1;i<=n;i++)if(f[i]==i)sum++;
    cout<<sum;
}
```

二分图最大匹配

匈牙利算法

```
#include<bits/stdc++.h>

using namespace std;
#define N 101
int e[N][N];
int match[N];
int book[N];
int n,m;
int dfs(int u){
    for(int i=1;i<=n;i++){
        if(book[i]==0&&e[u][i]==1){
            book[i]=1;
            if(match[i]==0||dfs(match[i])){
                match[i]=u;
                return 1;
            }
        }
    }
}
int main(){
    cin>>n>>m;
    int sum=0;
    int t1,t2;
    for(int i=1;i<=m;i++){
        cin>>t1>>t2;
        e[t1][t2]=1;
    }
    for(int i=1;i<=n;i++)match[i]=0;
    for(int i=1;i<=n;i++){
        for(int j=1;j<=n;j++)book[j]=0;//清除上次搜索时的标记
        if(dfs(i))sum++;//寻找增广路，找到配对数加1
    }
    cout<<sum<<endl;
}
```

堆

实现堆排序的两种方法

```
#include<bits/stdc++.h>

using namespace std;
#define N 101
```

```
int h[N];
int n;//记录堆的大小
int num;//记录数组大小
void swap(int a,int b){
    int t=h[a];
    h[a]=h[b];
    h[b]=t;
    return ;
}
void siftdown(int i){
    int t,flag=0;
    while(i*2<=n&&flag==0){
        if(h[i]<h[i*2])t=i*2;
        else t=i;

        if(i*2+1<=n&&h[t]<h[i*2+1])t=i*2+1;

        if(t!=i){
            swap(t,i);
            i=t;
        }
        else flag=1;
    }
    return;
}
void creat(){
    for(int i=n/2;i>=1;i--){
        siftdown(i);
    }
}

void heapsort(){
    while(n>1){
        swap(n,1);
        n--;
        siftdown(1);
    }
}
int main(){
    cin>>n;
    num=n;
    for(int i=1;i<=num;i++)cin>>h[i];
    creat();//建堆
    heapsort();
    for(int i=1;i<=num;i++)cout<<h[i]<<" ";
}
```

```
#include<bits/stdc++.h>

using namespace std;
#define N 101
int h[N];
int n;//记录堆的大小
int num;//记录数组大小
```

```cpp
void swap(int a,int b){
    int t=h[a];
    h[a]=h[b];
    h[b]=t;
    return ;
}
void siftdown(int i){
    int t,flag=0;
    while(i*2<=n&&flag==0){
        if(h[i]>h[i*2])t=i*2;
        else t=i;

        if(i*2+1<=n&&h[t]>h[i*2+1])t=i*2+1;

        if(t!=i){
            swap(t,i);
            i=t;
        }
        else flag=1;
    }
    return;
}
void creat(){
    for(int i=n/2;i>=1;i--){
        siftdown(i);
    }
}

int deletemax(){
    int t=h[1];
    h[1]=h[n];
    n--;
    siftdown(1);
    return t;
}
int main(){
    cin>>n;
    num=n;
    for(int i=1;i<=num;i++)cin>>h[i];
    creat();//建堆

    for(int i=1;i<=num;i++)cout<<deletemax()<<" ";
}
```