Lời giải bài DIJKSTRA

Bài này chỉ cần dùng thuật toán Dijkstra một cách bình thường nhất là sẽ giải được

Chú ý đồ thị có những cạnh một chiều và hai chiều, cần dựng đồ thị thật chính xác

Lời giải bài FLOYD

Tương tự bài trên, chỉ cần cài thuật toán Floyd bình thường là sẽ giải quyết được bài này

Cũng chú ý đến các cạnh một chiều và hai chiều của đồ thị

Lời giải bài MAXPATH

Bài toán yêu cầu với mỗi đỉnh X, gọi $d(1), d(2), ..., d(n)$ là độ dài đường đi ngắn nhất từ X đến các đỉnh đó, cần trả về giá trị lớn nhất của các giá trị d đó

Với $n \leq 100$, ta dùng Floyd để tính độ dài đường đi ngắn nhất giữa tất cả các cặp đỉnh

Thì kết quả với mỗi x là $max(d(x,1), ..., d(x,n))$

Lời giải bài TRIP

Với $n \leq 100$, ta dùng Floyd để tính độ dài đường đi ngắn nhất giữa tất cả các cặp đỉnh

Xuất phát từ 1, ta cứ làm theo yêu cầu đề bài: Tìm đỉnh x chưa được thăm có giá trị $d(1,x)$ nhỏ nhất, thăm x. Sau đó lại lấy x làm điểm xuất phát và làm tương tự như trên

Chú ý n phải được thăm cuối cùng, do đó khi lặp thì chỉ được phép lặp từ 2 đến $n-1$

Lời giải bài MAZE2

Coi mỗi ô là một đỉnh. Từ đỉnh (x,y) đến đỉnh kề (x',y') sẽ có trọng số là $a(x,y)$

Dùng thuật toán Dijkstra để tìm đường đi ngắn nhất

Chú ý tính cả bước nhảy ra ngoài mê cung

Giả mã:

```
typedef pair<int,int> ii;
typedef pair<long long,ii> li;

int n,m;
long long d[][];
int mark[][];
priority_queue<li,vector<li>,greater<li>> pq;

d[][]=+oo, mark[][]=0;
d[sx][sy]=0, pq.push({d[sx][sy],{sx,sy}});
```

```
while(pq.size()>0){
    {d[x][y],{x,y}}=pq.top();pq.pop();
    if (mark[x][y]==0) continue;
    mark[x][y]=1;
    for(){
        (xx,yy) kề với (x,y)
        if (d[x][y]+a[x][y]<d[xx][yy]){
            d[xx][yy]=d[x][y]+a[x][y];
            pq.push({d[xx][yy],{xx,yy}})
        }
    }
}
```

Lời giải bài CABLE

Gọi $d1(s,u)$ là độ dài đường đi ngắn nhất từ s tới u, $d2(t,v)$ là độ dài đường đi ngắn nhất từ t tới v

Để tính $d1(s,u)$ dùng Dijkstra bình thường, còn tính $d2(t,v)$ cần đảo chiều đồ thị mới có thể Dijkstra đúng được

Trước khi bổ sung cáp, độ dài đường đi ngắn nhất từ s đến t là $d1(s,t)$ (hoặc $d2(t,s)$)

Khi bổ sung đoạn cáp (u,v), đường đi ngắn nhất từ s đến t nếu qua đoạn cáp này có thể là:

- $d1(s,u)+w(u,v)+d2(t,v)$ (Đi theo chiều $u \to v$)
- $d1(s,v)+w(v,u)+d2(t,u)$ (Đi theo chiều $v \to u$)

Ta sẽ lấy min của tất cả những giá trị này để tính kết quả