



C++ 算法

深度优先搜索

Depth-first Search

例题：前m名

世界杯已进入8强争霸，输入m和8个队名，输出所有前m名的搭配可能性，每一行输出一种可能性:先冠军依次到第m名。行和行之间按照字典序排列。

输入样例

2

Mexico
Germany
Brazil
Argentina
China
Spain
Russia
France

输出样例

Argentina Brazil
Argentina China
Argentina France
Argentina Germany
Argentina Mexico
Argentina Russia
Argentina Spain
Brazil Argentina

.....

以下省略若干行

输入样例

3

Mexico
Germany
Brazil
Argentina
China
Spain
Russia
France

输出样例

Argentina Brazil China
Argentina Brazil France
Argentina Brazil Germany
Argentina Brazil Mexico
Argentina Brazil Russia
Argentina Brazil Spain
Argentina China Brazil

.....

以下省略若干行

如何枚举所有搭配

两种代码比较

```
12 void dfs(int x){ ←
13     if(x==m){print();return;}
14     for(int i=0;i<n;i++)
15         if(!vst[i]){
16             vst[i]=1;p[x]=i;
17             dfs(x+1);
18             vst[i]=0;
19         }
20 }
21 int main(){
22     cin>>m;
23     for(int i=0;i<n;i++)
24         cin>>team[i];
25     sort(team,team+n);
26     dfs(0);
27     return 0;
28 }
```

```
12 int x; ←
13 void dfs(){ ←
14     if(x==m){print();return;}
15     for(int i=0;i<n;i++)
16         if(!vst[i]){
17             vst[i]=1;p[x]=i;x++;
18             dfs();
19             vst[i]=0;x--;
20         }
21 }
22 int main(){
23     cin>>m;
24     for(int i=0;i<n;i++)
25         cin>>team[i];
26     sort(team,team+n);
27     x=0; ←
28     dfs(); ←
29     return 0;
30 }
```

状态转移

方法1 “显性”

利用**dfs()**函数的参数记录状态
通过调用**dfs()**时
设置不同参数进行状态转移

方法2 “隐性”

利用全局变量记录状态
需要在调用**dfs()**前后
自行维护状态转移

例题：最短路四方向有障碍版

在一个 $n*m$ 格子的迷宫里，o代表空地可以行走，#代表墙体不可以通过。从左上角(1,1)开始，每一步可以上下左右四个方向走动一格，请问需要至少几步可以达到(n,m)， $n,m \leq 10$

输入样例：

2 2

oo

#o

输出样例：

2

输入样例：

4 5

o#ooo

o#o#o

o#o#o

ooo#o

输出样例：

13

输入样例：

5 3

ooo

o#o

ooo

o##

ooo


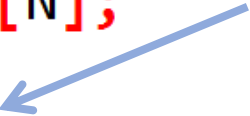
输出样例：

6

思考：如何打印出一条最短路

路径打印

```
8 int n,m,dist[N][N],pre[N][N];
9 char f[N][N];
10 void dfs(int x,int y,int z,int len){
11     if(len>=dist[x][y])return;
12     dist[x][y]=len; pre[x][y]=z;
13     if(x==n&&y==m)return;
14     for(int k=0;k<4;k++){
15         int nx=x+dx[k],ny=y+dy[k];
16         if(nx>=1&&nx<=n&&ny>=1&&ny<=m&&f[nx][ny]=='o')
17             dfs(nx,ny,k,len+1);
18     }
19 }
```



主函数调用
dfs(1,1,-1,0)

路径打印

```
20 void print(int x,int y){  
21     int k=pre[x][y]; ←  
22     if(k!=-1){  
23         int nx=x-dx[k],ny=y-dy[k];  
24         print(nx,ny); ←  
25     }  
26     cout<<x<<" "<<y<<endl;  
27 }
```


综合练习