

# C++编程

# DFS算法应用

## 枚举全排列

快快编程  
kkcoding.net

# 冠亚军

世界杯已进入4强争霸，输入4个队名，输出**所有冠亚军搭配的可能性**，每一行输出一种可能性:先冠军再亚军。行和行之间按照字典序排列。

输入样例：

Brazil  
Germany  
Spain  
Argentina

输出样例：

Argentina Brazil  
Argentina Germany  
Argentina Spain  
Brazil Argentina  
Brazil Germany  
Brazil Spain  
Germany Argentina  
Germany Brazil  
Germany Spain  
Spain Argentina  
Spain Brazil  
Spain Germany



排列

顺序有关

组合

顺序无关

如何枚举所有排列



快快编程  
kkcoding.net

# 循环枚举

team[i]代表i号球队队名

```
4  int n=4;
5  string team[10];
6  for(int i=1;i<=n;i++)cin>>team[i];
7  sort(team+1,team+1+n); ←
8  for(int i=1;i<=n;i++)
9      for(int j=1;j<=n;j++)
10         if(i!=j)
11             cout<<team[i]<<" "<<team[j]<<endl;
```

枚举冠军为i号球队

枚举亚军为j号球队



# 冠亚季军

世界杯已进入4强争霸，输入4个队名，输出所有冠亚季军搭配的可能性，每一行输出一种可能性：先冠军，亚军，再季军。行和行之间按照字典序排列。

如何枚举所有排列

输入样例：

Brazil  
Germany  
Spain  
Argentina

输出样例：

Argentina Brazil Germany  
Argentina Brazil Spain  
Argentina Germany Brazil  
Argentina Germany Spain  
Argentina Spain Brazil  
Argentina Spain Germany  
Brazil Argentina Germany  
Brazil Argentina Spain  
Brazil Germany Argentina  
Brazil Germany Spain  
Brazil Spain Argentina  
Brazil Spain Germany  
Germany Argentina Brazil  
Germany Argentina Spain  
Germany Brazil Argentina  
Germany Brazil Spain  
Germany Spain Argentina  
Germany Spain Brazil  
Spain Argentina Brazil  
Spain Argentina Germany  
Spain Brazil Argentina  
Spain Brazil Germany  
Spain Germany Argentina  
Spain Germany Brazil





# 冠亚季军

team[i]代表i号球队队名

```
int n=4;
string team[10];
for(int i=1;i<=n;i++)cin>>team[i];
sort(team+1,team+1+n);
for(int i=1;i<=n;i++)
    for(int j=1;j<=n;j++)if(i!=j)
        for(int k=1;k<=n;k++)if(k!=i&& k!=j)
            cout<<team[i]<<" "<<team[j]<<" "<<team[k]<<endl;
```



# 前m名

世界杯已进入n强争霸，输入n和m，以及n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。  $m \leq n \leq 10$

# 枚举排列

共有 $n$ 个数 $\{1,2,3,4,5,6,7,8\}$ ，从中选 $m$ 个不同的数进行排列。  
按照字典序输出所有排列。 $m \leq n \leq 10$

输入： 3 3	输出什么？
	输出： 1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

for循环层数  
控制不方便

解决方案  
用递归形式枚举

dfs(x)枚举决策：  
x号排位选数字几  
x号排位选哪支球队

dfs(x)递归调用  
dfs(x+1)



vst[i]代表i号球队是否被选过

visited缩写为vst

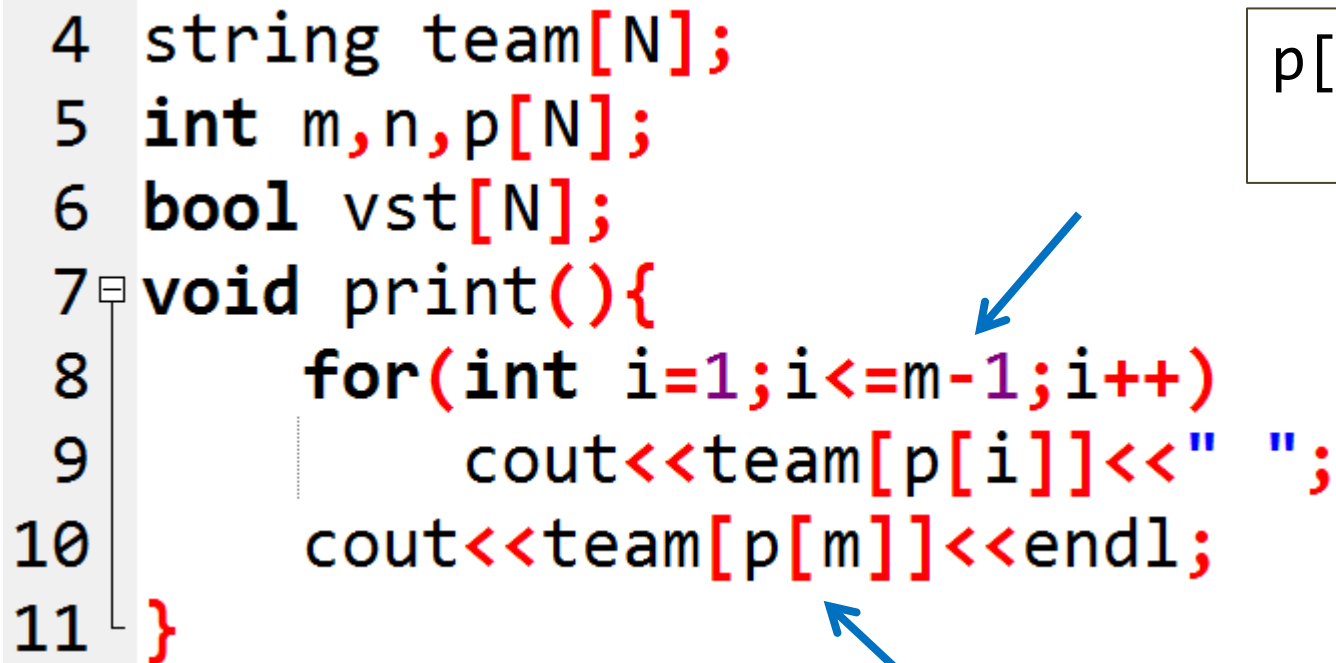
n代表球队总数

m代表需要选出几个排名

team[i]代表i号  
球队队名

p[x]代表x号排位  
选几号球队

```
4 string team[N];
5 int m,n,p[N];
6 bool vst[N];
7 void print(){
8     for(int i=1;i<=m-1;i++)
9         cout<<team[p[i]]<<" ";
10    cout<<team[p[m]]<<endl;
11 }
```



确保行末没有空格

vst[i]代表i号球队是否被选过

visited缩写为vst

n代表球队总数

m代表需要选出几个排名

team[i]代表i号  
球队队名

p[x]代表x号排位  
选几号球队

```
4 string team[N];
5 int m,n,p[N];
6 bool vst[N];
7 void print(){
8     for(int i=1;i<=[ ];i++)
9         cout<<[ ]<<" ";
10    cout<<team[p[m]]<<endl;
11 }
```

补全  
程序

dfs(x)枚举决策：  
x号排位选哪支球队

vst[i]代表i号球队是否被选过  
visited缩写为vst

```
12 void dfs(int x){
13     if(x==m+1){print();return;}
14     for(int i=1;i<=n;i++){
15         if(!vst[i]){
16             vst[i]=1;p[x]=i;
17             dfs(x+1);
18             vst[i]=0;
19         }
20     }
```

枚举x号排位选i号球队

若i号球队没被选过

p[x]代表x号排位  
选几号球队

```
22 cin>>n>>m;
23 for(int i=1;i<=n;i++)cin>>team[i];
24 sort( );
25 dfs( );
```

补全  
程序

# 方法对比

## 循环多层枚举

for循环枚举1号位

for循环枚举2号位

for循环枚举3号位

for循环枚举4号位

.....

## dfs多层枚举

dfs(1)内循环枚举1号位

dfs(2)内循环枚举2号位

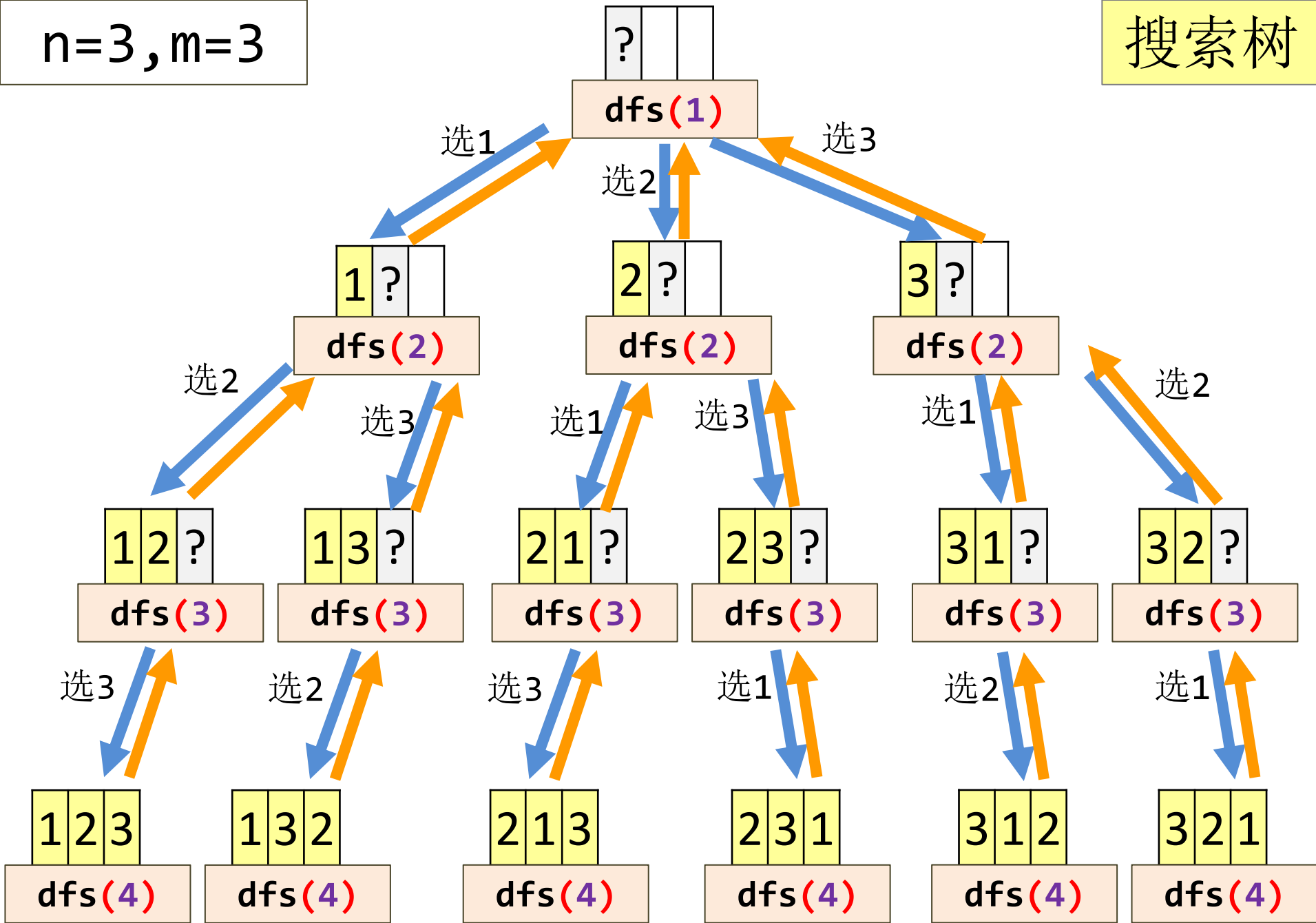
dfs(3)内循环枚举3号位

dfs(4)内循环枚举4号位

.....

$n=3, m=3$

搜索树



# DFS算法应用

## 枚举决策

决策: 还不确定的  
有待决定的信息

连续2格  
不同色

# 染色

需要对一排 $n$ 个格子进行染色。每个格子只可以是RGB这3种颜色之一。要求**相邻两个格子不能同色**，输入 $n$ ，输出所有可能性，每行一种染色方法，按照字典序排列。 $n \leq 10$

输入：	输出：
3	BGB BGR BRB BRG GBG GBR GRB GRG RBG RBR RGB RGR

决策 每格填什么颜色

$\text{dfs}(x)$ 枚举决策：  
 $x$ 号格子选哪个颜色

$\text{dfs}(x)$ 递归调用  
 $\text{dfs}(x+1)$



## dfs多层枚举

dfs(1)内循环枚举1号格子选哪个颜色

dfs(2)内循环枚举2号格子选哪个颜色

dfs(3)内循环枚举3号格子选哪个颜色

dfs(4)内循环枚举4号格子选哪个颜色

.....

枚举颜色时要保证  
连续2格不同色

n代表格子总数

colors[i]代表  
i号颜色的符号

```
3  const int N=20;
4  const int CLR=3;
5  char colors[CLR]={'B','G','R'};
6  int n,p[N];
7  void print(){
8      for(int i=1;i<=n;i++)
9          cout<<colors[p[i]];
10     cout<<endl;
11 }
```

p[x]代表x号格子  
选几号颜色

dfs(x)枚举决策：  
x号格子选哪个颜色

```
12 void dfs(int x){
13     if(x==n+1){print();return;}
14     for(int i=0;i<CLR;i++){
15         → if(x==1 || p[x-1]!=i){
16             p[x]=i;
17             dfs(x+1);
18         }
19     }
```

递归终止条件

枚举x号格子选i号颜色

若x号格子在最左边  
或者x-1号格子已选  
颜色不是i号颜色

p[x]代表x号格子  
选几号颜色

```
20 int main(){
21     cin>>n;
22     dfs(1);
23     return 0;
24 }
```

连续3格  
不同色

# 染色

需要对一排 $n$ 个格子进行染色。每个格子只可以是ABCD这4种颜色之一。要求**连续3格不能出现同色**。  
输入 $n$ ，输出所有可能性，每行一种染色方法，按照字典序排列。 $n \leq 10$

决策 每格填什么颜色

输入：  
3

$\text{dfs}(x)$ 枚举决策：  
 $x$ 号格子选哪个颜色

$\text{dfs}(x)$ 递归调用  
 $\text{dfs}(x+1)$

输出：  
ABC  
ABD  
ACB  
ACD  
ADB  
ADC  
BAC  
BAD  
BCA  
BCD  
BDA  
BDC  
CAB  
CAD  
CBA  
CBD  
CDA  
CDB  
DAB  
DAC  
DBA  
DBC  
DCA  
DCB

## dfs多层枚举

dfs(1)内循环枚举1号格子选哪个颜色

dfs(2)内循环枚举2号格子选哪个颜色

dfs(3)内循环枚举3号格子选哪个颜色

dfs(4)内循环枚举4号格子选哪个颜色

.....

枚举颜色时要保证  
连续3格不同色

dfs(x)枚举决策：  
x号格子选哪个颜色

p[x]代表x号格子  
选几号颜色

递归终止条件

```
12 void dfs(int x){
13     if(x==n+1){print();return;}
14     for(int i=0;i<CLR;i++)
15         → if(x==1 || x==2&& p[1]!=i ||
16             x>2&& p[x-1]!=i&& p[x-2]!=i){
17         p[x]=i;
18         dfs(x+1);
19     }
20 }
```

可以在x号格子  
选i号颜色的条件

或者

或者

x号格子在最左边

x号格子在第2格并且  
1号格子已选颜色不是i号颜色

x大于第2格并且  
x-1号格子和x-2号格子  
已选颜色都不是i号颜色

连续2格  
不同色

# 有固定色的染色

要对一排格子进行染色，有些格子已有固定颜色。每个格子只可以是RGB这3种颜色之一。要求**相邻两格不能同色**，输入一串字符s代表格子情况，o代表空格RGB代表已有固定色。输出所有可能性，每行一种染色方法，按照字典序排列。n<=10

输入：	输出：
RoR	RBR
	RGR

输入：	输出：
ooRG	BGRG
	GBRG
	RBRG
	RGRG

输入：	输出：
oBo	GBG
	GBR
	RBG
	RBR

dfs(x)枚举决策：  
x号格子选哪个颜色

dfs(x)递归调用  
dfs(x+1)

决策	每一格填什么
----	--------

编程  
kkcoding.net



## dfs多层枚举

dfs(1)内循环枚举1号格子选哪个颜色

dfs(2)内循环枚举2号格子选哪个颜色

dfs(3)内循环枚举3号格子选哪个颜色

dfs(4)内循环枚举4号格子选哪个颜色

.....

枚举颜色时要保证  
连续2格不同色  
已有固定颜色不可以改变

键盘输入时用  
Ctrl Z结束

不断输入字符

```
23 int main(){
24     char ch;
25     while(cin>>ch){
26         ++n;
27         if(ch=='B')p[n]=0;
28         else if(ch=='G')p[n]=1;
29         else if( ) 
30         else p[n]=-1;
31     }
32     dfs(1);
33     return 0;
34 }
```

dfs(x)枚举决策：  
x号格子选哪个颜色

p[x]代表x号格子  
选几号颜色

```
12 void dfs(int x){
13     if(x==n+1){print();return;}
14     if( ) {dfs(x+1);return;}
15     for(int i=0;i<CLR;i++)
16         if(
17             p[x]=i;
18             dfs(x+1);
19             p[x]=-1;
20         }
21 }
22 }
```

可以在x号格子  
选i号颜色的条件

`next_permutation()`

下一个

排列

参考链接: [http://www.cplusplus.com/reference/algorithm/next\\_permutation/](http://www.cplusplus.com/reference/algorithm/next_permutation/)

n=3

请预测  
输出结果

完成程序  
验证结果

头文件<algorithm>

```
cin>>n;  
for(int i=1;i<=n;i++)p[i]=i;  
do{  
    print();  
}while(next_permutation(p+1,p+1+n));
```

快快网  
kkcoding.net

# 快快编程作业

518

519

734

拓展题

932