

# C++编程

# 搜索算法

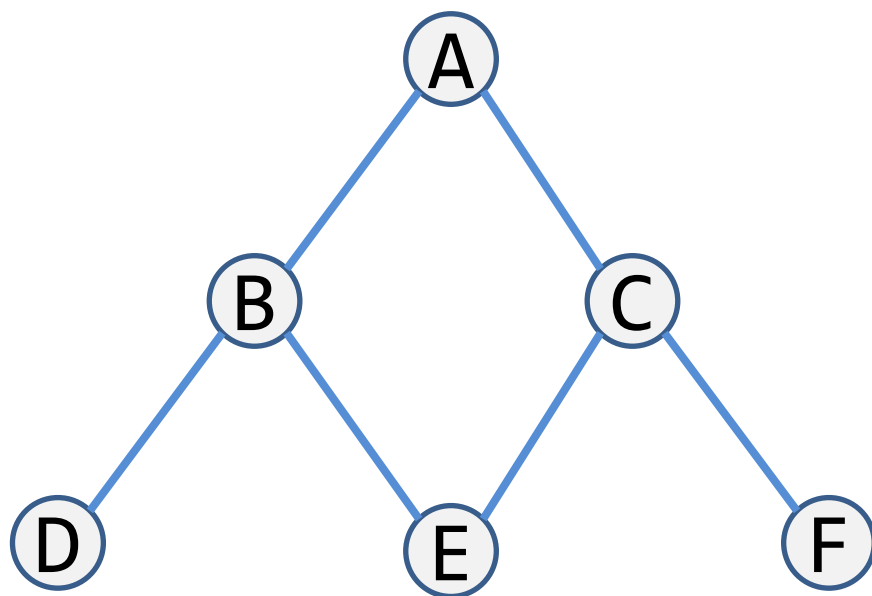
一种枚举方法



快快编程  
kkcoding.net

# 藏宝图 DFS

你拿到一张地下藏宝图，标有若干藏宝洞。你从A洞出发进行搜索，访问这几个洞的顺序会是怎么样的呢？



ABDECF

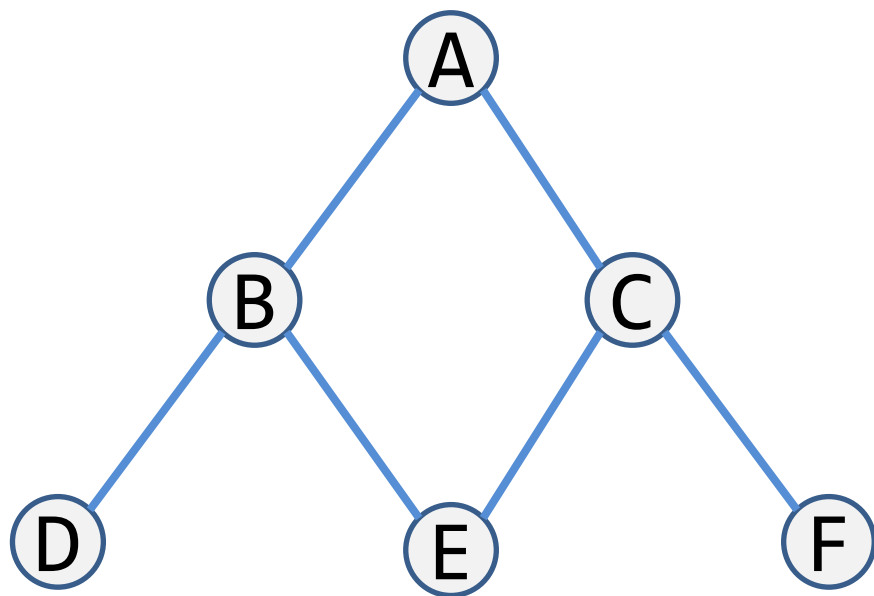
深度优先搜索

depth-first search

简称DFS

# 藏宝图 BFS

你拿到一张地下藏宝图，标有若干藏宝洞。你从A洞放了一把火，火势蔓延开，这几个洞被火烧的顺序会是怎样的呢？



**ABCDEF**

广度优先搜索

**breadth-first search**

简称BFS

DFS算法应用

枚举子集

# 自助餐

作为大胃王，你最爱吃自助餐，享受大快朵颐的满足感。

已知自助餐有 $n$ 种菜品，每种菜你都可以选择吃或不吃。当然啦，你最少会吃1种，最多会吃 $n$ 种。请输出所有可能性。

输入样例：

```
3
baozi
longxia
niupai
```

输出样例：

```
baozi longxia niupai
baozi longxia
baozi niupai
baozi
longxia niupai
longxia
niupai
```

暗中观察



# 枚举全部子集

$n \leq 15$

有 $n$ 个数字组成的集合， $\{1, 2, 3, \dots, n\}$ 。请枚举所有非空子集。

输出的先后顺序遵循以下规则：

1. 每行输出一个子集，每个子集的元素从小到大输出，由空格隔开，行末不能有空格。
2. 较小数字开头的子集比较大数字开头的子集先输出。
3. 开头数字一样的话，再依次比较后续数字。后续有数字的子集先输出，后续没有数字的子集后输出。

输入样例： 输出样例：

3

1 2 3

1 2

1 3

1

2 3

2

3

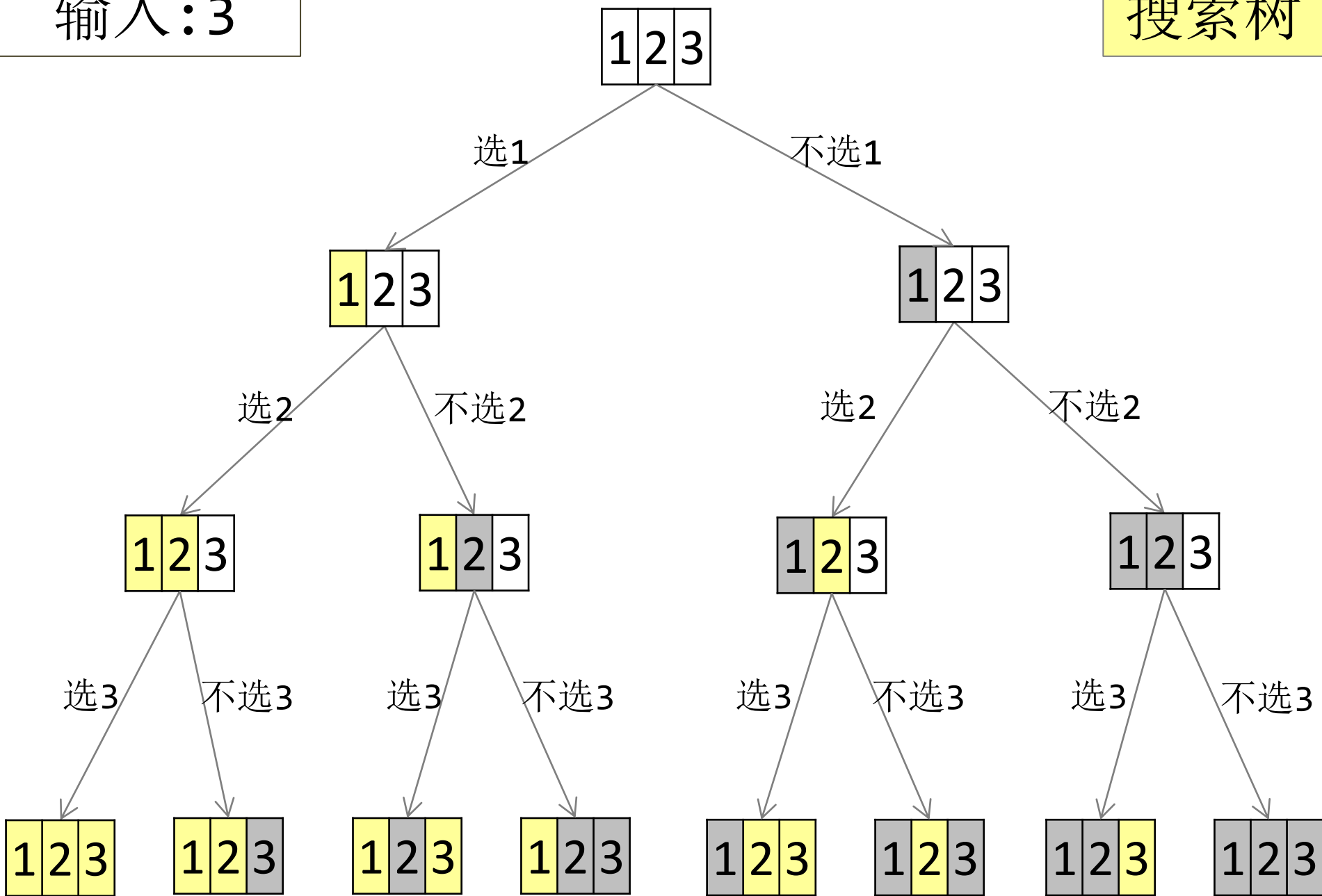
非空"子集"包含  
所有大小的"组合"

没有个数限制

总共几种方案？

输入:3

搜索树





# 枚举算法

共有 $n$ 个数字要考虑

依次枚举每个数字

对每个数做**决策** 选或不选

程序代码这么写?

ok[i]代表  
数字i是否选择

p[i]代表  
第i位选几

```
3 int n,ok[20],p[20];
4 void print(){
5     int m=0;
6     for(int i=1;i<=n;i++)
7         if(ok[i])p[++m]=i;
8     if(m==0)return;
9     for(int i=1;i<m;i++)
10         cout<<p[i]<<" ";
11     cout<<p[m]<<endl;
12 }
```

确保行末无空格

**for**循环层数  
控制不方便

解决方案  
用递归形式枚举

```

13 void dfs(int x){
14     if(x==n+1){
15         print();
16         return;
17     }
18     ok[x]=1;
19     dfs(x+1);
20     ok[x]=0;
21     dfs(x+1);
22 }

```

当前候选数字x

候选数字x越界

ok[i]代表  
数字i是否选择

决策	选x
----	----

决策	不选x
----	-----

主函数调用  
dfs(1);

ok[i]代表  
数字i是否选择

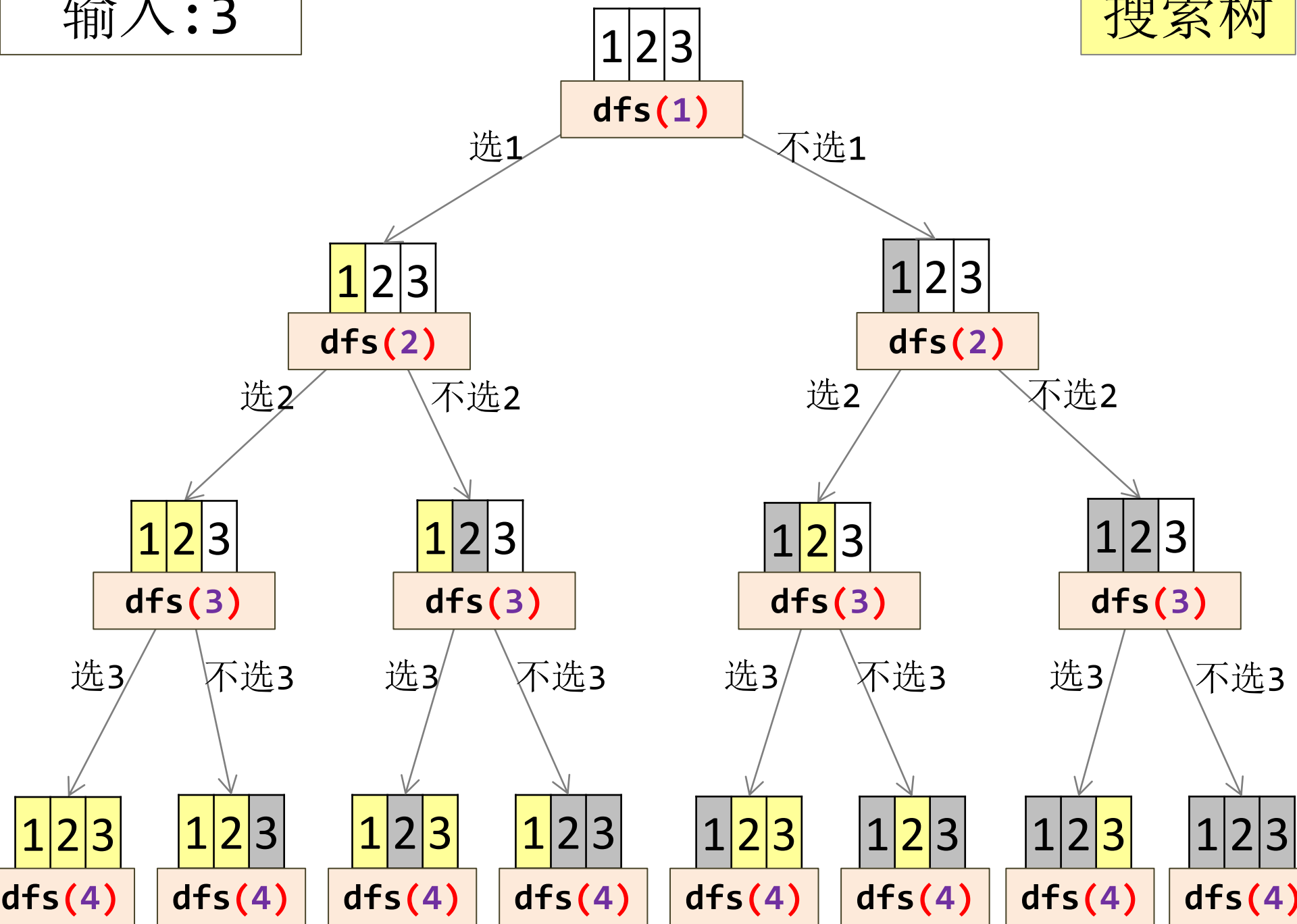
p[i]代表  
第i位选几

```
3 int n,ok[20],p[20];
4 void print(){
5     int m=0;
6     for(int i=1;i<=n;i++)
7         if(ok[i])p[++m]=i;
8     if(m==0)return;
9     for(int i=1;i<m;i++)
10         cout<<p[i]<<" ";
11     cout<<p[m]<<endl;
12 }
```

确保行末无空格

输入:3

搜索树



DFS算法应用

枚举子集

# 中奖名单

有 $n$ 个人参加你主办的幸运大抽奖活动，你决定随机选 $m$ 个人得奖， $1 \leq m \leq n$ 。请枚举所有可能方案。

输入样例：	输出样例：
3 2	Alice Bob
Alice	Alice Carl
Bob	Bob Carl
Carl	





# 枚举全部组合

有 $n$ 个数字组成的集合， $\{1, 2, 3, \dots, n\}$ 。给定一个 $m$ ， $1 \leq m \leq n$ 。对于从 $n$ 个数里选 $m$ 个数的组合情况，请枚举所有可能方案。

输出的先后顺序遵循以下规则：

1. 每行输出一个组合，每个组合的元素从小到大输出，由空格隔开，行末不能有空格。
2. 较小数字开头的组合比较大数字开头的组合先输出。
3. 开头数字一样的话，再依次比较后续数字。

输入样例：

3 2

输出样例：

1 2

1 3

2 3

注意："组合"不是"排列"

选出的数字顺序不重要

如 $\{2, 3\}$ 和 $\{3, 2\}$ 算同一种组合

# 两种枚举算法

1 依次决策每个数 选不选

是非题

2 依次决策每个位置 选哪个数

选择题

决策类型  
有区别

同学投票  
哪一种方法更好

**1**

依次决策每个数 选不选

是非题

快快编程  
kkcoding.net

# 1

依次决策每个数 选不选

是非题

当前候选数字x

已经选了c个数

```
9 void dfs(int x, int c){  
10     if(c==m){  
11         print();  
12         return;  
13     }  
14     if(x==n+1) return;  
15     p[c+1]=x;  
16     dfs(x+1, c+1);  
17     dfs(x+1, c);  
18 }
```

已选够m个数

候选数字x越界

p[i]代表第i位选几

决策

选x

决策

不选x

# 1 依次决策每个数 选不选

是非题

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int n,m,p[20];
4  void print(){
5      for(int i=1;i<m;i++)
6          cout<<p[i]<<" ";
7      cout<<p[m]<<endl;
8  }
9  void dfs(int x,int c){

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

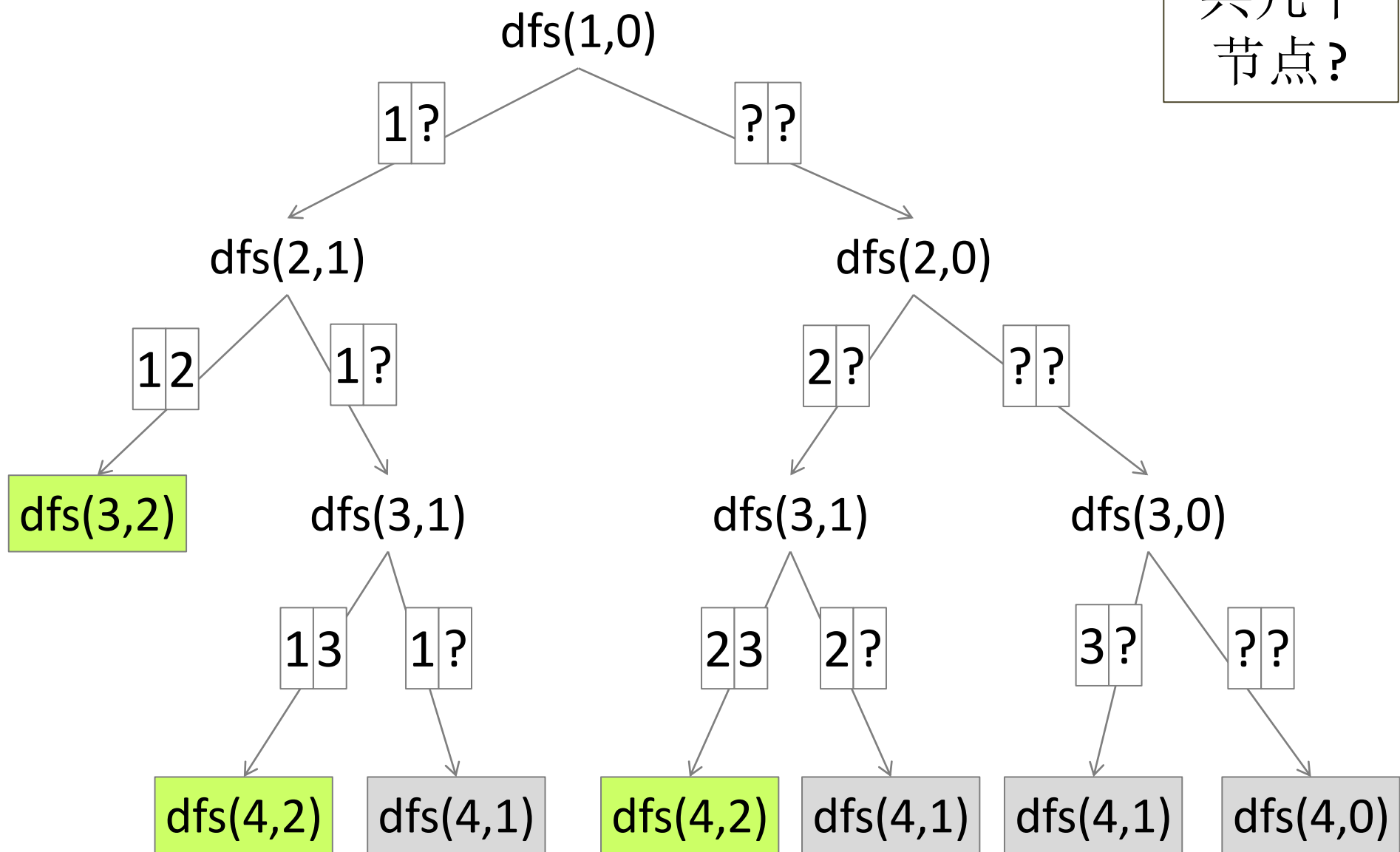
p[i]代表第i位选几  
确保行末无空格

枚举所有组合  
当前候选数字1  
已经选了0个数

输入: 3 2

搜索树

共几个  
节点?



**1**

依次决策每个数 选不选

是非题

优化加速

减少枚举量

# 1

依次决策每个数 选不选

是非题

当前候选数字x

已经选了c个数

```
9 void dfs(int x, int c){
10     if(c == m){
11         print();
12         return;
13     }
14     if(c + n + 1 - x < m) return;
15     if(x == n + 1) return;
16     p[c + 1] = x;
17     dfs(x + 1, c + 1);
18     dfs(x + 1, c);
19 }
```

可选数字  
还剩下  $n+1-x$  个

若已选数字加  
剩下可选数字  
都不够  $m$  个

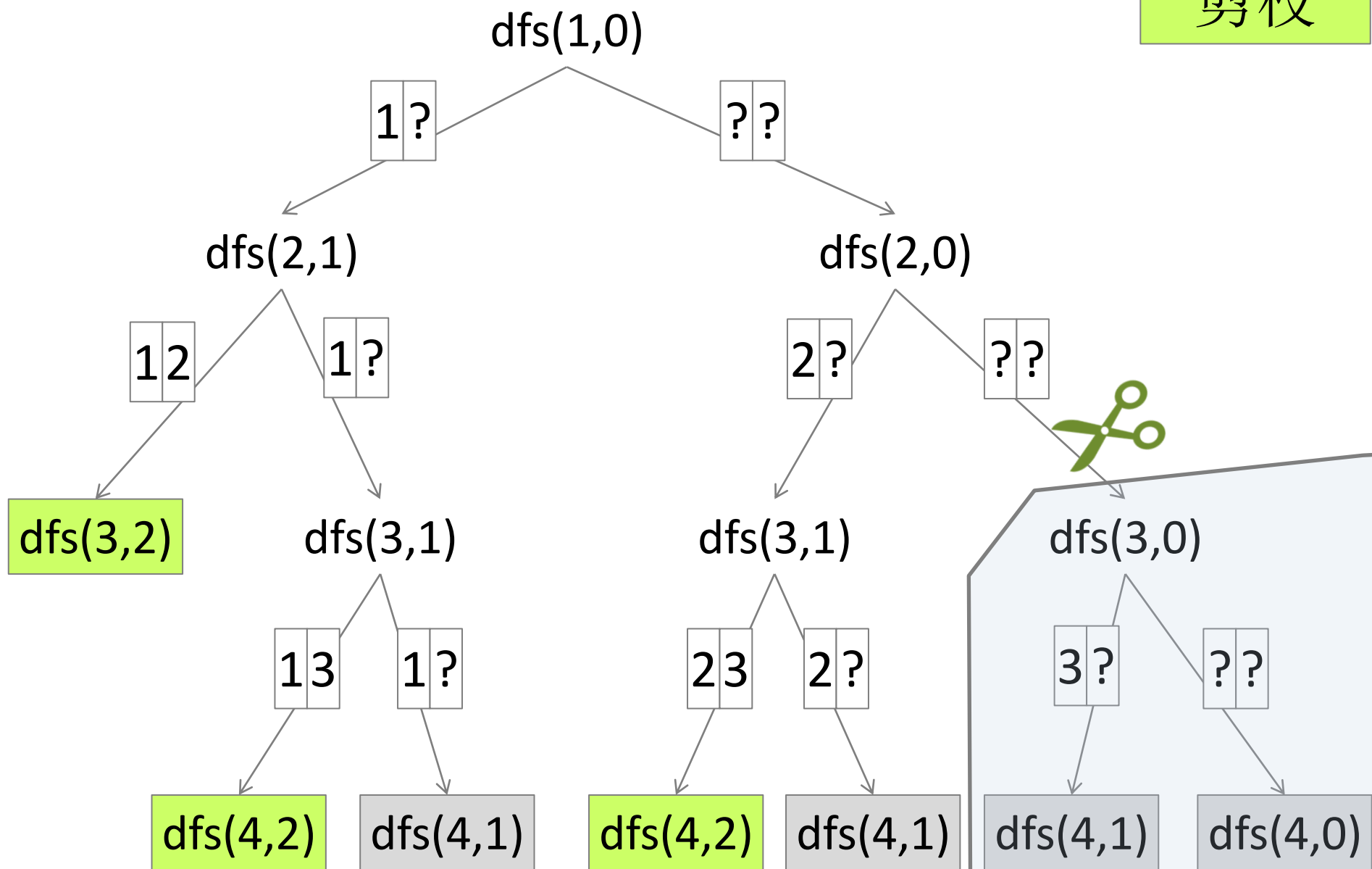
提前结束枚举



输入: 3 2

搜索树

剪枝



2

依次决策每个位置

选哪个数

选择题

快快编程  
kkcoding.net

## 2

依次决策每个位置

选哪个数

选择题

当前候选数字起点x

已经选了c个数

```
9 void dfs(int x, int c){
10     if(c==m){
11         print();
12         return;
13     }
14     for(int i=x; i<=n; i++){
15         p[c+1]=i;
16         dfs(i+1, c+1);
17     }
18 }
```

已选够m个数

枚举第c+1个数  
选哪个

决策

选中i

## 2

## 依次决策每个位置

## 选哪个数

## 选择题

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int n,m,p[20];
4 void print(){
5     for(int i=1;i<m;i++)
6         cout<<p[i]<<" ";
7     cout<<p[m]<<endl;
8 }
9 void dfs(int x,int c){
19 int main(){
20     cin>>n>>m;
21     dfs(1,0);
22     return 0;
23 }
```

确保行末无空格

枚举所有组合

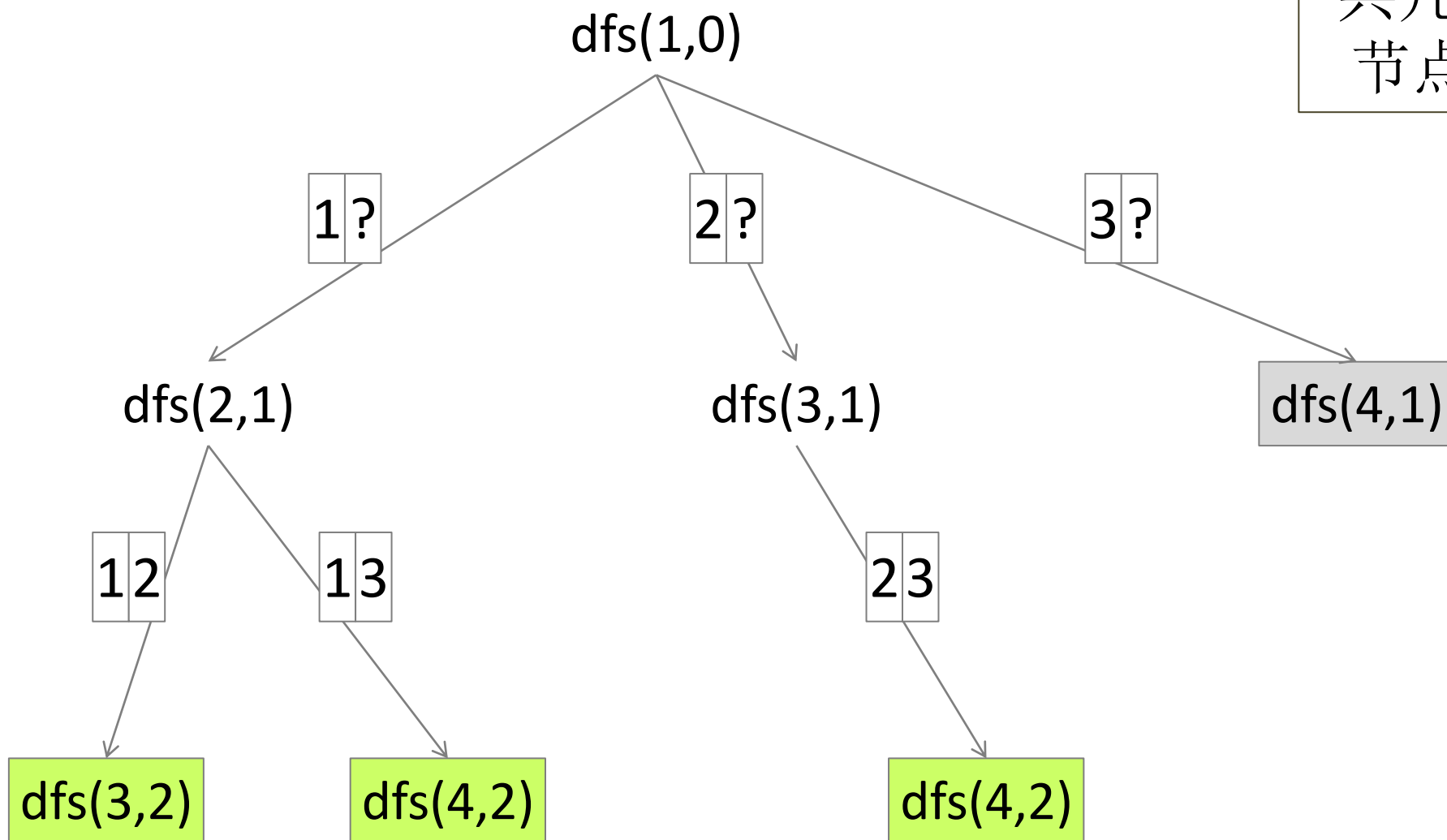
当前候选数字  
从1开始

已经选了0个数

输入: 3 2

搜索树

共几个  
节点?



2

依次决策每个位置

选哪个数

选择题

优化加速

减少枚举量

## 2

依次决策每个位置

选哪个数

选择题

当前候选数字起点x

已经选了c个数

```
9 void dfs(int x, int c){
10     if(c == m){
11         print();
12         return;
13     }
14     if(c + n + 1 - x < m) return;
15     for(int i = x; i <= n; i++){
16         p[c + 1] = i;
17         dfs(i + 1, c + 1);
18     }
19 }
```

可选数字  
还剩下  $n+1-x$  个

若已选数字加  
剩下可选数字  
都不够  $m$  个

提前结束枚举

## 2

依次决策每个位置

选哪个数

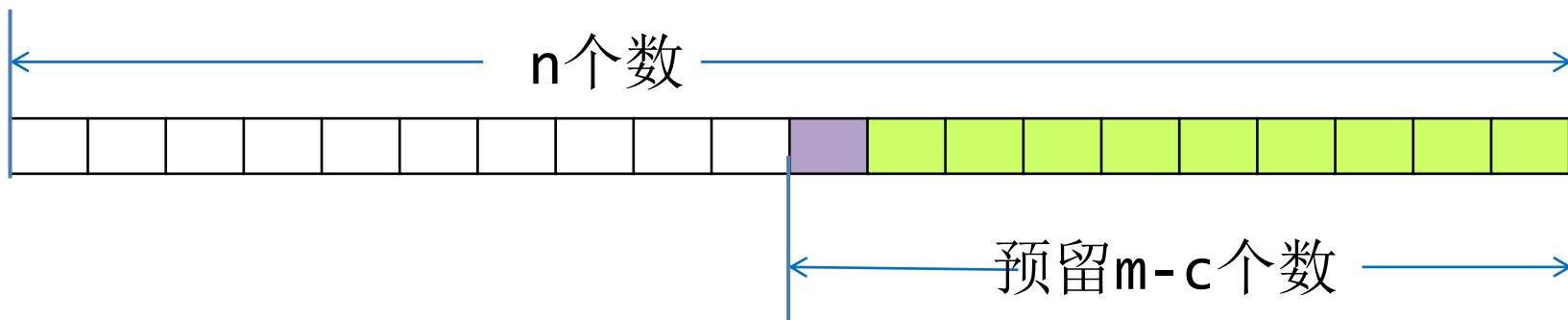
选择题

```
9 void dfs(int x, int c){
10     if(c == m){
11         print();
12         return;
13     }
14     for(int i = x; i <= n + 1 - m + c; i++){
15         p[c + 1] = i;
16         dfs(i + 1, c + 1);
17     }
18 }
```

减少枚举量

选数不能太靠后

边界数字  $n + 1 - m + c$

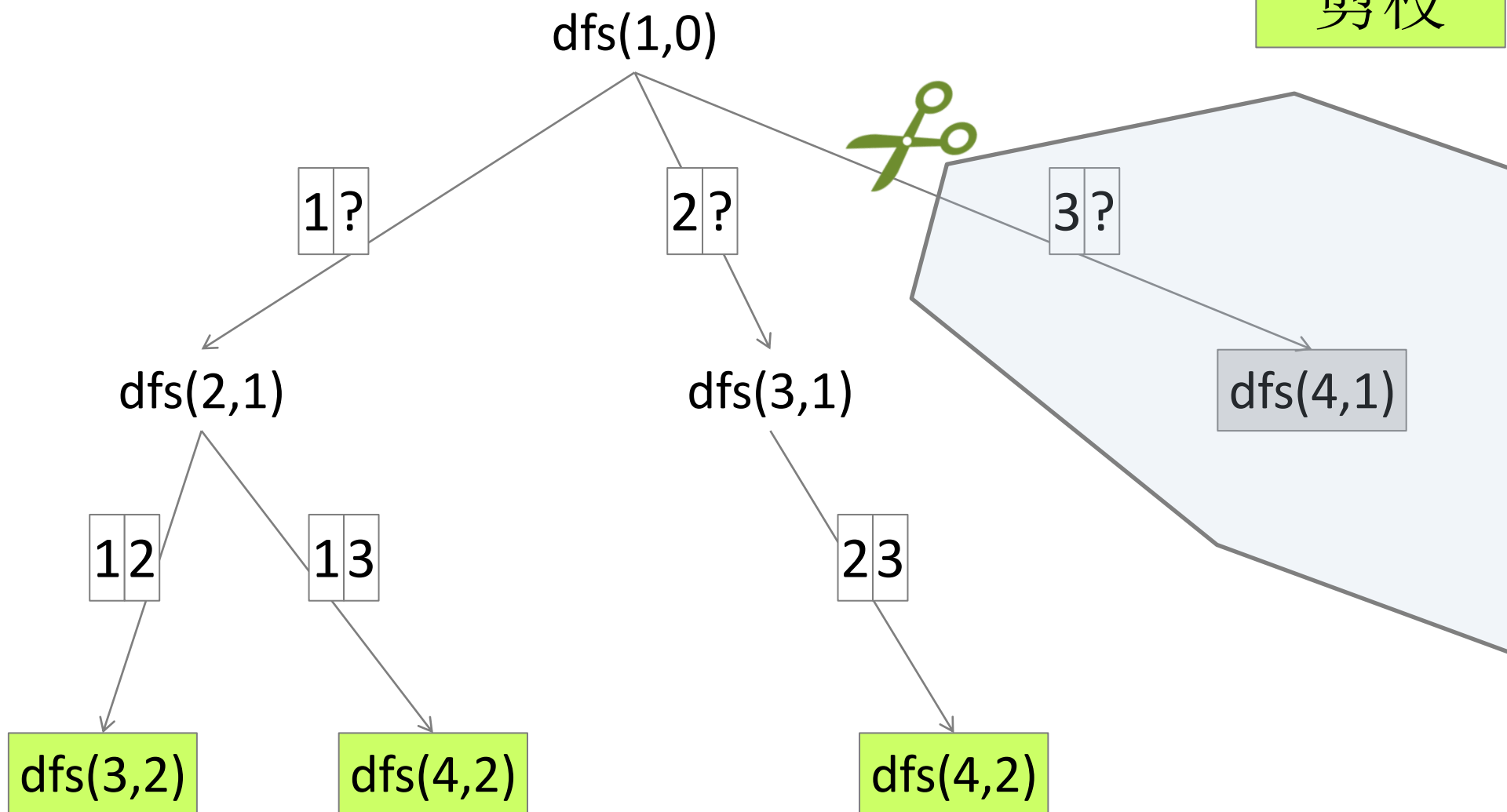




输入: 3 2

搜索树

剪枝



# 比较：两种枚举算法

1 依次决策每个数 选不选

2 依次决策每个位置 选哪个数

有哪些区别和优劣？

# 快快编程作业

1657. 枚举子集

1658. 枚举组合

拓展题

42. 选数

拓展题

297. 分解因数