

C++算法

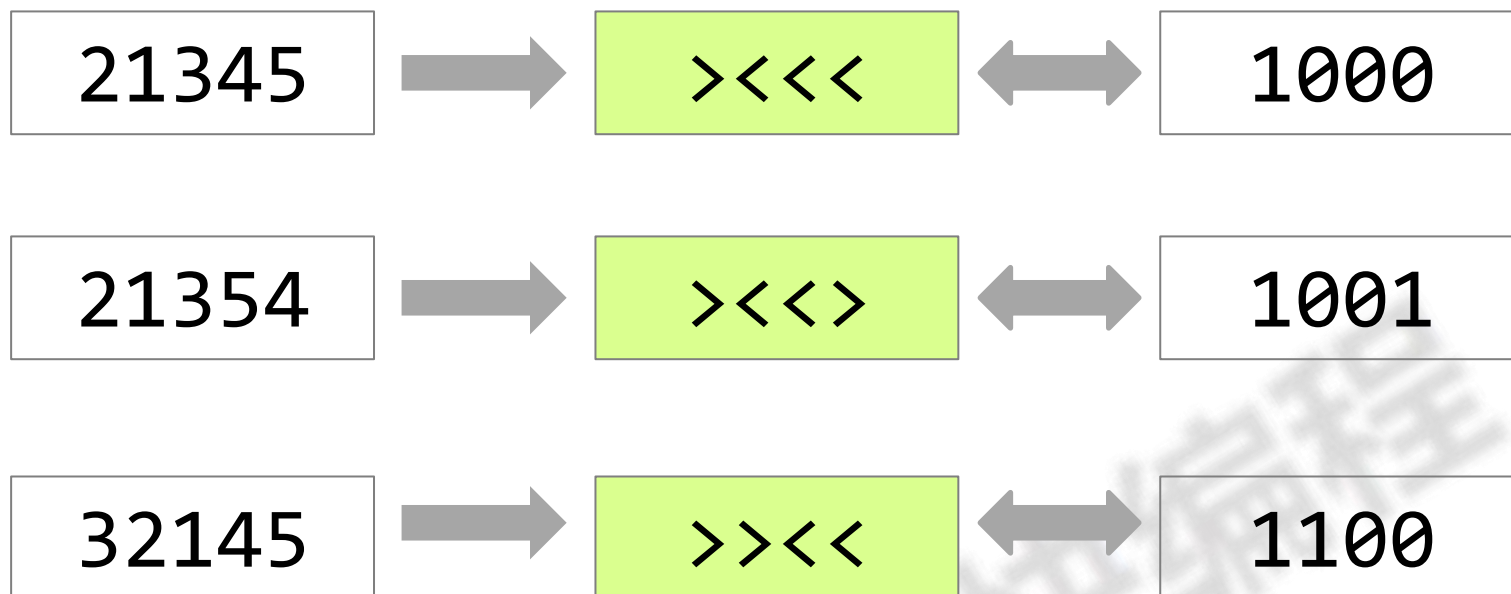


快快编程2170

快快编程
kkcoding.net



信息的01表示



两种思路

A

暴力枚举排列

B

动态规划



暴力枚举

枚举 $n!$ 种排列

对每种排列
判断是否合法?

快快编程
kkcoding.net



```
40 cin>>n;
41 for(ll i=1;i<=n;++i)cin>>a[i];
42 for(ll i=1;i<n;++i){
43     if(a[i]<a[i+1])s[i]='<';
44     else s[i]='>';
45 }
46 dfs(1);
47 cout<<ans<<endl;
```

快快编程
kkcoding.net



```
24 void dfs(ll x){
25     if(x==n+1){
26         if(valid())ans++;
27         return;
28     }
29     for(ll i=1;i<=n;++i){
30         if(ok[i]) 
31         p[x]=i;
32         ok[i]=1;
33         dfs(x+1);
34         
35     }
36 }
```

p[x]表示第x位排哪个数

ok[i]表示i是否使用过



暴力+优化

枚举排列时
能否提前筛选合法排序？

<><>

不需要考虑

21...

31...

41...

51...

32...

42...

43...

.....



```
47 ☐
48
49
50
51
52
53
for(ll i=1;i<=n;++i){
    p[1]=i;
    ok[i]=1;
    dfs(2);
    ok[i]=0;
}
cout<<ans<<endl;
```

能否直接调用
dfs(1)

快快编程
kkcoding.net



```
17 void dfs(ll x){
18     if(x==n+1){ans++;return;}
19     if( ) {
20         for(ll i=p[x-1]+1;i<=n;++i){
21             if(ok[i])continue;
22             p[x]=i;
23             ok[i]=1;
24             dfs(x+1);
25             ok[i]=0;
26         }
27     }
28     else{
29         for( ) {
30             if(ok[i])continue;
31             p[x]=i;
32             ok[i]=1;
33             dfs(x+1);
34             ok[i]=0;
35         }
36     }
37 }
```

p[x]表示第x位排哪个数

ok[i]表示i是否使用过

快快编程
kkcoding.net



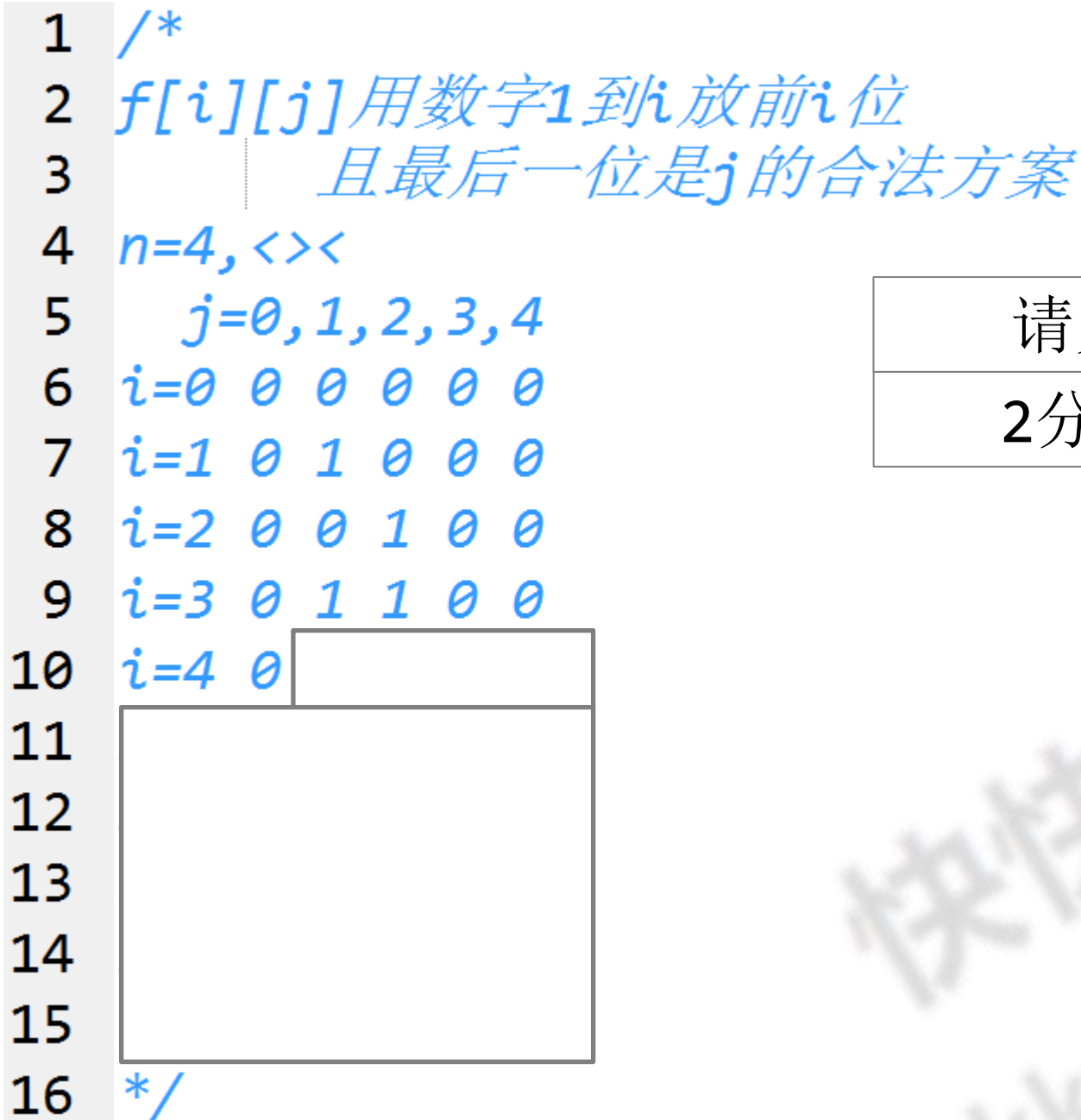
请同学写出状态定义

$f[n][m]$ 表示 1到n的排列中末尾是m的合法排列数量

$f[i][j]$ 表示什么含义?

$f[i][j]$ 表示 只考虑前i个位置,1到i的排列中末尾是j的合法排列数量

快快编程
kkcoding.net



2分钟后老师检查

f[i][j]用数字1到i放前i位且最后一位是j的合法方案

n=4,<><

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	0	0	0
i=1	0	1	0	0	0
i=2	0	0	1	0	0
i=3	0	1	1	0	0
i=4	0	0	1	2	2

a[3]<a[4]

f[4][4]依赖f[3][1]+f[3][2]+f[3][3]

因为a[4]为4时,a[3]可以是1,2,3

重大发现

每格依赖上一行的前缀和！

若对应a[i-1]<a[i]

f[i][j]

=

f[i-1][j-1]

+

f[i][j-1]



$n=4, <><$

若最后一位是小于号<
每格依赖上一行的前缀和!

$n=4, ><>$

若最后一位是大于号>
每格依赖上一行的后缀和?

快速编程
kkcoding.net

转移方程

$f[i][j]$ 用数字1到i放前i位且最后一位是j的合法方案

$n=4, ><>$

	$j=0$	$j=1$	$j=2$	$j=3$	$j=4$
$i=0$	0	0	0	0	0
$i=1$	0	1	0	0	0
$i=2$	0	1	0	0	0
$i=3$	0	0	1	1	0
$i=4$	0	2	2	1	0

$a[3]>a[4]$

$f[4][1]$ 依赖 $f[3][3]+f[3][2]+f[3][1]$

因为 $a[4]$ 为1时, $a[3]>a[4]$ 对应 $a[3]$ 为4,3,2

重大发现

每格依赖上一行的后缀和!

若对应 $a[i-1]>a[i]$

$f[i][j]$

=

$f[i-1][j]$

+

$f[i][j+1]$

```
33 f[1][1]=1;
34 for(ll i=2;i<=n;++i){
35     if(s[i-1]=='<'){
36         for(ll j=2;j<=i;++j)
37             f[i][j]=(f[i][j-1]+f[i-1][j-1])%MOD;
38     }
39     else{
40         for( )
41             f[i][j]=
42     }
43 }
```




快快编程2171

快快编程
kkcoding.net

两种思路

A

暴力枚举

B

动态规划



分组问题的枚举
如何设计枚举过程
枚举对象+枚举顺序

1

枚举分几批+枚举每人在哪批

2

枚举当前这批有谁

快速编程
kkcoding.net

两种思路

A

暴力枚举

B

动态规划



信息的01表示

$f[p]$ 表示让集合状态 p 都过桥至少几分钟

$f[7]=f[(111)_2]$ 表示让3个人
0号1号2号都过桥至少几分钟

$f[6]=f[(110)_2]$ 表示让2个人
1号2号都过桥至少几分钟

对于 n 答案
输出什么

$f[(1 \ll n) - 1]$



```
1  /*
2  f[p] 让集合状态p 都过桥至少几分钟
3  w=2, n=3,
4  t[0]=3, w[0]=1
5  t[1]=2, w[1]=1
6  t[2]=1, w[2]=1
7
8      p = 0      1      2      3      4      5      6      7
9          000  001  010  011  100  101  110  111
10 f[p]= 0      3      2      3      1      
11 */
```

请完成第1-10行

2分钟后老师检查



信息的01表示

预处理信息可反复调用

$sw[q]$ 集合状态 q 对应的人的体重总和

$mT[q]$ 集合状态 q 对应的人里最长过桥时间



```

19 cin>>W>>n;
20 for(int i=0;i<n;++i)cin>>t[i]>>w[i];
21 int nPtn=(1<<n);
22 for(int q=0;q<nPtn;++q){
23     for(int i=0;i<n;++i)if( ){
24         sW[q]+= ;
25         mT[q]=max( );
26     }
27 }

```




$f[p]$ 表示让集合状态 p 都过桥至少几分钟

转移决策

枚举一批
人选集合
用 q 标记

```
28 for(int p=1;p<nPtn;++p){
29     f[p]=
30     for(int q=1;q<nPtn;++q)if() {
31         if(sW[q]>W)continue;
32         f[p]=min(f[p],f[p-q]+mT[q]);
33     }
34 }
35 cout<<f[nPtn-1]<<endl;
```

能否改
 $f[p^q]$

如何
加速?

两种加速思路

A

预处理筛选合法状态

总重量 $\leq W$

B

只枚举 p 对应集合的子集 q

快快编程
kkcoding.net

A

预处理筛选合法状态

总重量 $\leq W$

```

22 int nOK=0;
23 for(int q=0;q<nPtn;++q){
24     for(int i=0;i<n;++i)if((1<<i)&q){
25         sW[q]+=w[i];
26         mT[q]=max(mT[q],t[i]);
27     }
28     if(sW[q]>W)continue;
29     ptn[nOK++]=q;
30 }

```

A

预处理筛选合法状态

总重量 $\leq W$

```

31  for(int p=1;p<nPtn;++p){
32      f[p]=INF;
33      for(int u=0;u<nOK;++u){
34          int q=ptn[u];
35          if( )continue;
36          f[p]=min(f[p],f[p-q]+mT[q]);
37      }
38  }
    
```

两种加速思路

A

预处理筛选合法状态

总重量 $\leq W$

B

只枚举 p 对应集合的子集 q

快快编程
kkcoding.net

加速
方案

mask
蒙版

B 只枚举p对应集合的子集q

例如 $p=6=(1010)_2$

3号	2号	1号	0号
可以选	不可选	可以选	不可选

1010

-1

1 0

1000

-1

0 1

0010

-1

0 0

0000

$(q-1) \& p$



mask
蒙版

kkcoding.net

29

30

31

32

33

34

35

```
for(int p=1;p<nPtn;++p){  
    f[p]=INF;  
    for(int q=p;q;q=(q-1)&p){  
        if(sW[q]>W)continue;  
        f[p]=min(f[p],f[p-q]+mT[q]);  
    }  
}
```

	参考：二项式定理
时间复杂度 $O(3^n)$	$\sum_{k=0}^n C(n,k) \times 2^k = (1+2)^n = 3^n$

快快编程作业

2170

2171

要求：先尝试暴力再用DP