



浙江财经大学

Zhejiang University Of Finance & Economics



2019初赛讲解

信智学院 陈琰宏

1 单项选择题

1. 中国的国家顶级域名是 (A)

A. cn

B. ch

C. chn

D. china

2. 二进制数11 1011 1001 0111 和01 0110 1110 1011进行逻辑与运算的结果是 (D)。

A. 01 0010 1000 1011

B. 01 0010 1001 0011

C. 01 0010 1000 0001

D. 01 0010 1000 0011

11 1011 1001 0111

01 0110 1110 1011

01 0010 1000 0011

3. 一个32位整型变量占用 (C) 个字节。

A. 32

B. 128

C. 4

D. 8

1 单项选择题

4. 若有如下程序段，其中s、a、b、c均已定义为整型变量，且a、c均已赋值(c大于0)

```
s=a;
```

```
for (b=1;b<=c;b++) s=s-1;
```

则与上述程序段功能等价的赋值语句是 (A)

A. **$s=a-c$;** B. $s=a-b$; C. $s=s-c$; D. $s=b-c$;

for循环b从1~c, 实现减去s-c, s的初始值为a, 所以实现 $a-c$

5. 设有100个已排好序的数据元素，采用折半查找时，最大比较次数为 (A)

A. **7** B. 10 C. 6 D. 8

$2^6 \leq 100 \leq 2^7$, 所以最大7次就可以

6. 链表不具有的特点是 (D)

A. 插入删除不需要移动元素

B. 不必事先估计存储空间

C. 所需空间与线性表长度成正比

D. 可随机访问任一元素

1 单项选择题

7. 把8个同样的球放在5个同样的袋子里，允许有的袋子空着不放，问共有多少种不同的分法？（ C ）提示：如果8个球都放在一个袋子里，无论是哪个袋子，都只算同一种分法

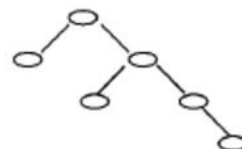
A. 22 B. 24 **C. 18** D. 20

枚举

0 0 0 0 8	0 0 0 4 4	0 0 2 2 4	0 1 1 3 3	1 1 1 2 3
0 0 0 1 7	0 0 1 1 6	0 0 2 3 3	0 1 2 2 3	1 1 2 2 2
0 0 0 2 6	0 0 1 2 5	0 1 1 1 5	0 2 2 2 2	
0 0 0 3 5	0 0 1 3 4	0 1 1 2 4	1 1 1 1 4	

8. 一棵二叉树如右图所示，若采用顺序存储结构，即用一维数组元素存储该二叉树中的结点(根结点的下标为1, 若某结点的下标为 i , 则其左孩子位于下标 $2i$ 处、右孩子位于下标 $2i+1$ 处)，则该数组的最大下标至少为（ C ）。

A. 6 B. 10 **C. 15** D. 12



1 单项选择题

9. 100 以内最大的素数是 (B) 。

A. 89 **B. 97** C. 91 D. 93

10. 319和377的最大公约数是 (C) 。

A. 27 B. 33 **C. 29** D. 31

根据答案凑就好。319和377都不能被3整除

11. 新学期开学了，小胖想减肥，健身教练给小胖制定了两个训练方案。
方案一：每次连续跑3公里可以消耗300千卡(耗时半小时)；方案二：每次连续跑.5公里可以消耗600千卡(耗时1小时)。小胖每周周一到周四能抽出半小时跑步，周五到周日能抽出一小时跑步。另外，教练建议小胖每周最多跑21 公里，否则会损伤膝盖。请问如果小胖想严格执行教练的训练方案，并且不想损伤膝盖，每周最多通过跑步消耗多少千卡？
(C)

A. 3000 B. 2500 **C. 2400** D. 2520

连续跑5公里肯定是更好的，所以 就在周五到周日跑5公里，周一到周四中选2天跑3公里。

1 单项选择题

12. 一副纸牌除掉大小王有52张牌，四种花色，每种花色13张。假设从这52张牌中随机抽取13张纸牌，则至少（ A ）张牌的花色一致。

A. 4 B. 2 C. 3 D. 5

四种花色各取1张，4张4张取， $4 \times 3 = 12$ ，还剩1张，所以必然有4张花色一样。

13. 一些数字可以颠倒过来看，例如0、1、8颠倒过来还是本身，6颠倒过来是9，9颠倒过来看还是6，其他数字颠倒过来都不构成数字。类似的，一些多位数也可以颠倒过来看，比如106颠倒过来是901。假设某个城市的车牌只由5位数字组成，每一位都可以取0到9。请问这个城市最多有多少个车牌倒过来恰好还是原来的车牌？（ C ）

A. 60 B. 125 C. 75 D. 100

 x y z y z

因为倒过来要跟原来的一样，车牌必然是xyzy形式，x, y可以在0, 1, 8, 6, 9中选择；z可以在0, 1, 8中选择。按照分步骤的乘法原理
 $5 \times 5 \times 3 = 75$

1 单项选择题

14. 假设一棵二叉树的后序遍历序列为DGJHEBIFCA, 中序遍历序列为DBGEHJACIF,

则其前序遍历序列为 (B) 。

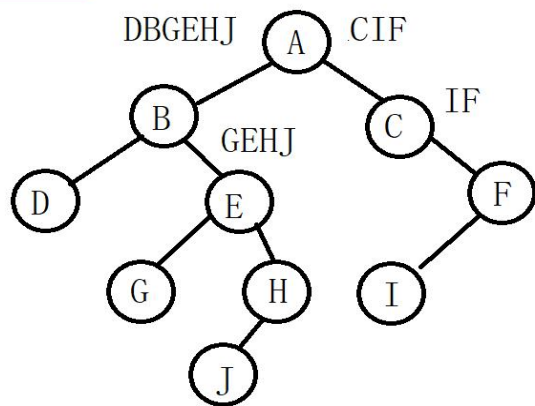
A. ABCDEFGHIJ

B. ABDEGHJCFI

C. ABDEGJHCFI

D. ABDEGHJFIC

后序DGJHEBIFCA, 中序遍历序列为DBGEHJACIF,



15 . 以下哪个奖项是计算机科学领域的最高奖? (A)

A. 图灵奖

B. 鲁班奖

C. 诺贝尔奖

D. 普利策奖

2 阅读程序写结果1[6567]

```
1  #include <stdio>
2  #include <string>
3  using namespace std;
4  char st[100];
5  int main() {
6      scanf( "%s", st);
7      int n = strlen(st);
8      for(int i=1; i<=n; ++i) {
9          if(n%i==0) {//如果字符串长度如果能整除当前字符的下标
10             char c=st[i-1];//就把字符小写变大写
11             if (c >= 'a')
12                 st[i - 1] = c-'a'+'A';
13             }
14     }
15     printf("%s", st);
16     return 0;
17 }
```


2 阅读程序写结果1

●判断题

- 1) 输入的字符串只能由小写字母或大写字母组成。 (F)
- 2) 若将第8行的“ $i=1$ ”改为“ $i=0$ ”，程序运行时会发生错误。 (T)

出现除0

- 3) 若将第8行的“ $i \leq n$ ”改为“ $i * i \leq n$ ”，程序运行结果不会改变。 (F)
- 4) 若输入的字符串全部由大写字母组成，那么输出的字符串就跟输入的字符串一样。 (T) 题目小写字母可能改大写，其它不会变

●选择题

- 5) 若输入的字符串长度为18，那么输入的字符串跟输出的字符串相比，至多有 (B) 个字符不同。 $n \% i == 0$, 才有机会改变, 18的约数有6个
A. 18 B. 6 C. 10 D. 1
- 6) 若输入的字符串长度为 (B) ，那么输入的字符串跟输出的字符串相比，至多有36个字符不同。求哪个数的因子有36个
A. 36 B. 100000 C. 1 D. 128

$100000 = 2^5 * 5^5$ 因子个数 = $(5+1) * (5+1) = 36$

2 阅读程序写结果2[6335]

```
1 #include <stdio>
2 using namespace std;
3 int n,m;
4 int a[100], b[100];
5
6 int main( ) {
7     scanf( "%d%d",&n, &m);
8     for(int i=1; i<=n; ++i)
9         a[i] = b[i] = 0;
10    for(int i=1; i<=m; ++i) {
11        int x, y;
12        scanf("%d%d", &x, &y);
13        if(a[x]<y&b[y]<x) {
14            if (a[x] > 0)
15                b[a[x]] = 0;
16            if (b[y] > 0)
17                a[b[y]] = 0;
18            a[x] = y;
19            b[y] = x;
20        }
21    }
```

```
22
23
24
25
26
27
28
29
30
31 }
int ans = 0;
for(int i=1; i<=n; ++i) {
    if (a[i] == 0)
        ++ans ;
    if (b[i] == 0)
        ++ans ;
}
printf("%d\n",ans);
return 0;
```

2 阅读程序写结果2[6335]

题目一看很抽象，不知道在做什么，可以 考虑直接模拟。
通过边界数据、举反例、特殊数据的模拟求解答案。选择题
可以找规律确定答案。

假设输入的n和m都是正整数，
x和y都是在[1, n]的范围内的整数，完
成下面的判断题和单选题：

●判断题.

1)当 $m > 0$ 时，输出的值一定小于 $2n$ 。（ T ）

由23-28行可推出ans的可能最大值一定 $\leq 2n$ 。题目初始， $[i]/b[i]$ 都等于0，
如果假设 $m=1$,输入x,y，因为x和y都是在[1, n]的范围内的整数所以13行必然
成立，则 $a[x] > 0$, $b[y] > 0$.推而广之， $a[x], b[y]$ 必然存在一个 > 0 的值。

2)执行完第27 行的 “++ans”时，ans 一定是偶数。（ F ）

3) $a[i]$ 和 $b[i]$ 不可能同时大于0。（ F ）

输入 $m=1, x=1, y=1$, 则 $a[1]=1, b[1]=1$

4)若程序执行到第13行时，x总是小于y,那么第15行不会被执行。（ F ）

15行不会被执行的前提 $a[x] > 0$ 。 $a[x] < y \& b[y] < x$, $x < y$, 没法推出 $a[x] \leq 0$

```
6 int main() {
7     scanf("%d", &n, &m);
8     for(int i=1; i<=n; ++i)
9         a[i] = b[i] = 0;
10    for(int i=1; i<=m; ++i) {
11        int x, y;
12        scanf("%d%d", &x, &y);
13        if(a[x]<y&b[y]<x) {
14            if (a[x] > 0)
15                b[a[x]] = 0;
16            if (b[y] > 0)
17                a[b[y]] = 0;
18            a[x] = y;
19            b[y] = x;
20        }
21    }
22    int ans = 0;
23    for(int i=1; i<=n; ++i) {
24        if (a[i] == 0)
25            ++ans;
26        if (b[i] == 0)
27            ++ans;
28    }
29    printf("%d\n", ans);
30    return 0;
31 }
```

5)若m个x两两不同, 且m个y两两不同, 则输出的值为 ()

A. $2n-2m$ B. $2n+2$ C. $2n-2$ D. $2n$

设 $m=1, x=1, y=2$ $a[1]=2, b[2]=1, ans=2$, 选项跟m没有关系, 显然是不对的, 所以选A

6)若m个x两两不同, 且m个y都相等, 则输出的值为 ()

A. $2n-2$ B. $2n$ C. $2m$ D. $2n-2m$

设 $m=2, x=1, y=1$ $a[1]=2, b[2]=1, ans=2$, 选项跟m没有关系, 显然是不对的, 所以选A

```
6 int main( ) {
7     scanf( "%d%d",&n, &m);
8     for(int i=1; i<=n; ++i)
9         a[i] = b[i] = 0;
10    for(int i=1; i<=m; ++i) {
11        int x, y;
12        scanf("%d%d", &x, &y);
13        if(a[x]<y&b[y]<x) {
14            if (a[x] > 0)
15                b[a[x]] = 0;
16            if (b[y] > 0)
17                a[b[y]] = 0;
18            a[x] = y;
19            b[y] = x;
20        }
21    }
```

```
22
23    int ans = 0;
24    for(int i=1; i<=n; ++i) {
25        if (a[i] == 0)
26            ++ans ;
27        if (b[i] == 0)
28            ++ans ;
29    }
30    printf("%d\n",ans);
31    return 0;
}
```

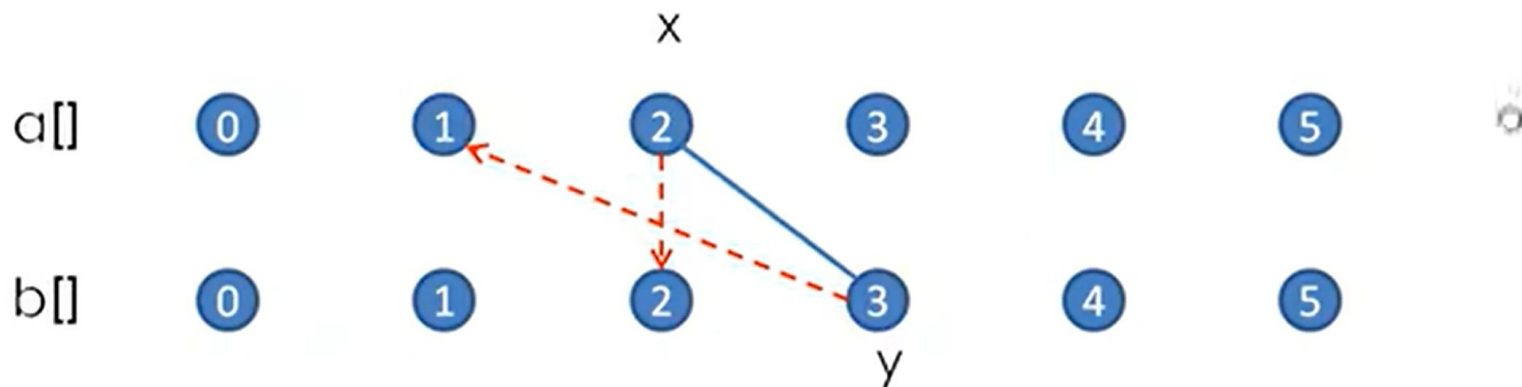
2 阅读程序写结果2[6335]

程序可以视作通过模拟算法选出一系列互不冲突的数对——若数对 (x_1, y_1) 和 (x_2, y_2) 之间有 $x_1 == x_2$ 或 $y_1 == y_2$ 则认为这两个数对存在冲突，现按顺序考虑每一个数对 (x_i, y_i) （要求满足前提：在已经选用的数对中，与左值 x_i 匹配的右值小于 y_i 、与右值 y_i 匹配的左值小于 x_i ），若该数对与此前已经选用的数对冲突，则用当前数对替换所冲突的原数对；若无冲突，则直接选用当前数对。程序中的 $a[x]$ 用于记录，在已选用的数对中，与左值 x 相匹配的右值； $b[y]$ 用于记录，在已选用的数对中，与右值 y 相匹配的左值； n 表示数对左值和右值的取值范围为 $[1, n]$ ；最后的 ans 用于统计剩余多少左值或右值，没有相应数对被我们选中。

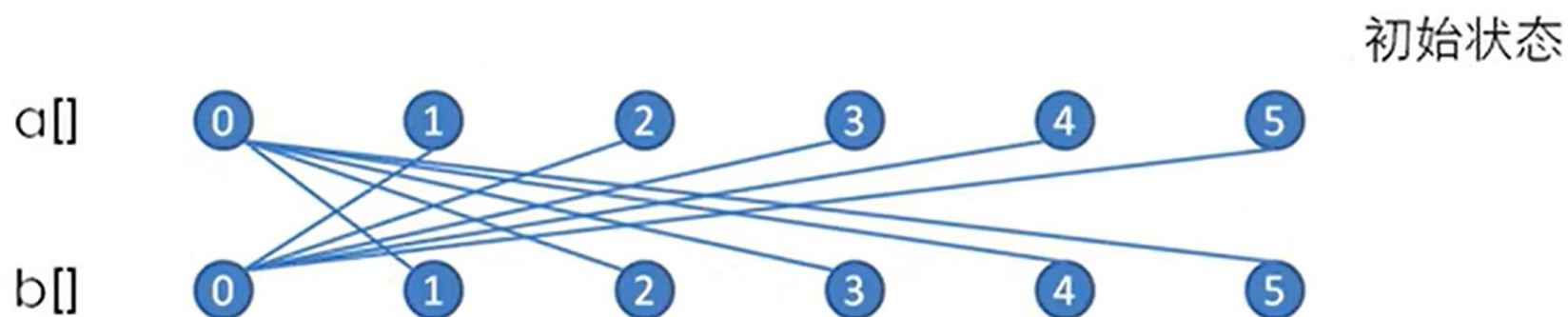
详细理解

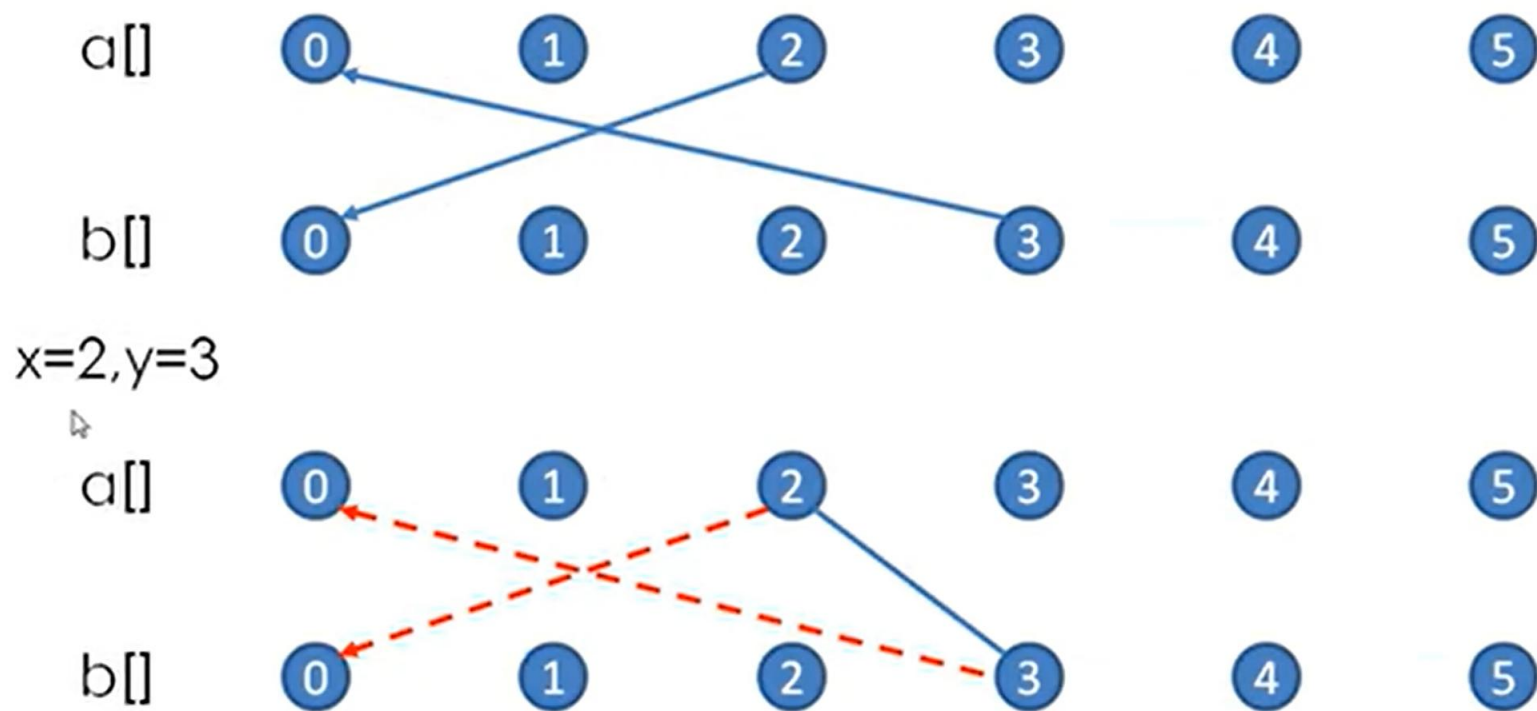
题目一看很抽象，不知道在做什么，可以 考虑直接模拟。通过边界数据、举反例、特殊数据的模拟求解答案。选择题可以找规律确定答案。

如果新的配对方案使与 $a[x]$ 和 $b[y]$ 配对的两个端点都向右移动（这两个元素之前的配对交叉了），就重新配对。蓝线为新配对方案，红线为原配对方案

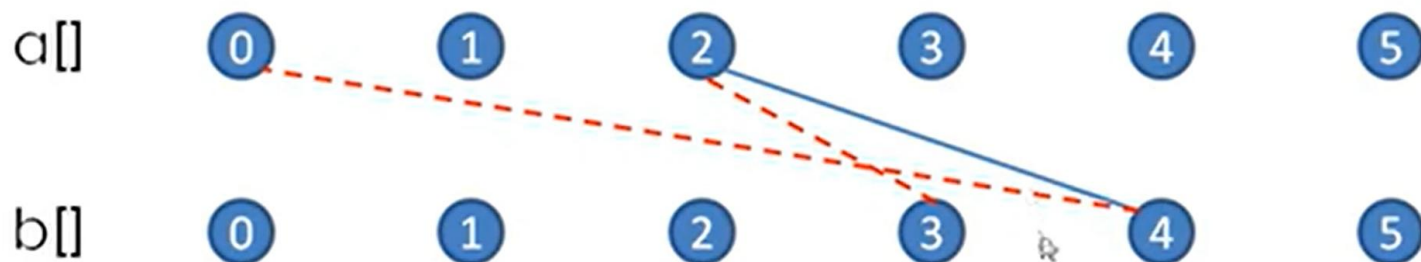


程序分析: 有a、b两组元素，两组元素之间相互两两对应。初始时所有元素都和另一组的0对应。每次考察a组的第x个元素和b组的第y个元素，如果新的配对方案使与a[x]和b[y]配对的两个端点都向右移动（这两个元素之前的配对交叉了），就重新配对。输出未配对元素数量。

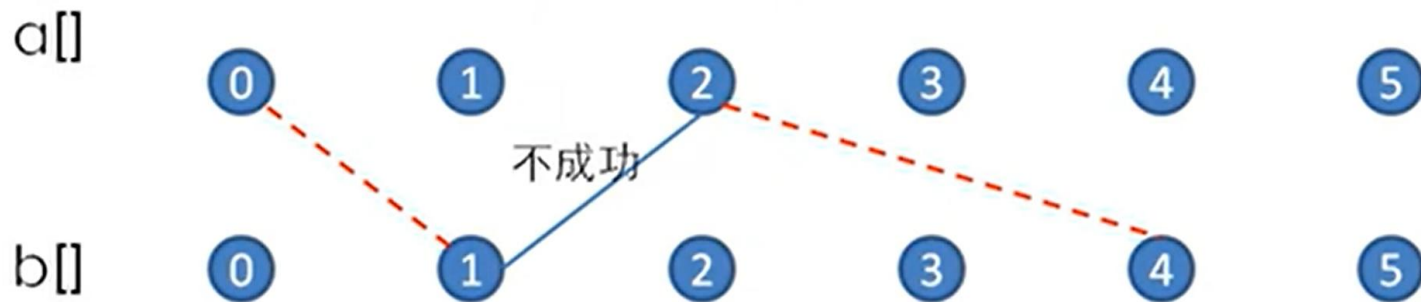




接上页，第2个数据， $x=2, y=4$ ， $a[2]$ 和 $b[4]$ 重新配对

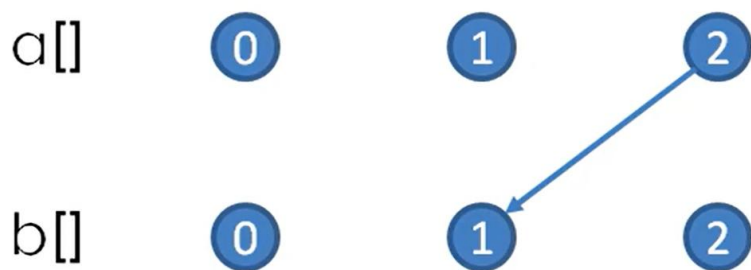


接上，第3个数据， $x=2, y=1$ ，无变化



-
- 1)当 $m>0$ 时，输出的值一定小于 $2n$ 。（ T ）
2)执行完第27行的“++ans”时，ans一定是偶数。（ F ）

本题相当于两个数组间有若干条线有对照关系，最终的ans必定为偶数，但是在for循环的统计过程中不一定。如下： $i=1$ 时，执行完27行，ans是奇数



```
22  int ans = 0;
23  for(int i=1; i<=n; ++i) {
24      if (a[i] == 0)
25          ++ans ;
26      if (b[i] == 0)
27          ++ans ;
28  }
29  printf("%d\n",ans);
30  return 0;
31 }
```

1)当 $m>0$ 时，输出的值一定小于 $2n$ 。（ T ）

2)执行完第27行的“++ans”时，ans一定是偶数。（ F ）

3) $a[i]$ 和 $b[i]$ 不可能同时大于0。（ F ）

输入 $m=1, x=1, y=1$, 则 $a[1]=1, b[1]=1$

4)若程序执行到第13行时， x 总是小于 y ,那么第15行不会被执行。（ F ）

15行不会被执行的前提 $a[x] > 0$ 。 $a[x] < y \& b[y] < x, x < y$ ，没法推出 $a[x] \leq 0$

5)若 m 个 x 两两不同, 且 m 个 y 两两不同, 则输出的值为 (**A**)

A. $2n-2m$ B. $2n+2$ C. $2n-2$ D. $2n$

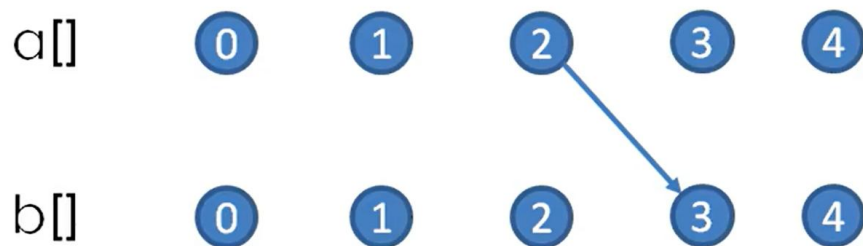
因为 m 个 x 和 m 个 y 互不相同, 会成功配对 m 次 (配对前各自初始值为0, 配对后不会更新)。 m 次循环中会有 $2m$ 个位置的值会变化, $ans=2n-2m$ 。也可举例: $m=2$, $x=1, y=2, x=3, y=4$

6)若 m 个 x 两两不同, 且 m 个 y 都相等, 则输出的值为 ()

A. $2n-2$ B. $2n$ C. $2m$ D. $2n-2m$

x 和 y 配对时每次都会清理掉之前的配对 (或者不更新配对), 最终只会产生一组配对, 所以和0配对的元素数量为 $2n - 2$ 。

举例: 第一组 $x=2, y=3$ 配对成功后, 若 $x=1, y=3$ 不更新配对, $x=3, y=3$ 更新原有配对。始终只有一对配对成功。



2 阅读程序写结果2

● 判断题

(1) 删除第15行程序, 运行结果不会发生变化。 (✓)

全部变量不赋值也等于0

(2) 将第16行中的i=1改为i=0, 运行结果不变。 (✓)

int i=0 就是多比较了一趟, 但结果不变

(3) 在17、18行中间插入 “if (get(i+i)==get(ans+i)) continue;”
运行结果不变。 (✓)

==时, 程序不做事情, 所以没关系

(4) 将21行的break换成continue, 因为这是for语句的最后一句, 所以
运行结果不变。 (✗)

改变了程序逻辑

2 阅读程序写结果3 [6343]二叉树求最小值

```
1  #include <iostream>
2  using namespace std;
3  const int maxn = 10000;
4  int n;
5  int a[maxn];
6  int b[maxn];
7  int f(int l, int r, int depth) {
8      if(l>r)
9          return 0;
10     int min = maxn, mink;
11     for(int i=l; i<=r; ++i) {
12         if (min > a[i]) {
13             min = a[i];
14             mink = i;
15         }
16     }
17     int lres = f(l, mink - 1, depth + 1);
18     int rres = f(mink + 1, r, depth + 1);
19     return lres + rres + depth * b[mink];
20 }
21 int main( ) {
22     cin >> n;
23     for(int i=0; i<n; ++i)
24         cin>> a[i];
25     for(int i=0; i<n; ++i)
26         cin>> b[i];
27     cout<<f(0,n-1,1)<<"\n";
28     return 0;
29 }
```

3 阅读程序写结果3

本题的含义为：输入n个整数 a_i 和n个整数 b_i ， a_i 与 b_i 的下标相等，一一对应。

根据 a_i 的值建立一颗二叉树，每次把当前区间内的最小值mink作为根，mink左边的元素作为左子树，mink右边的元素作为右子树，预定根节点的层级为1，求出每个节点的层级 h_i 。

最后求出每个元素层级 h_i 与当前元素 a_i 对于的 b_i 的乘积积和。即求

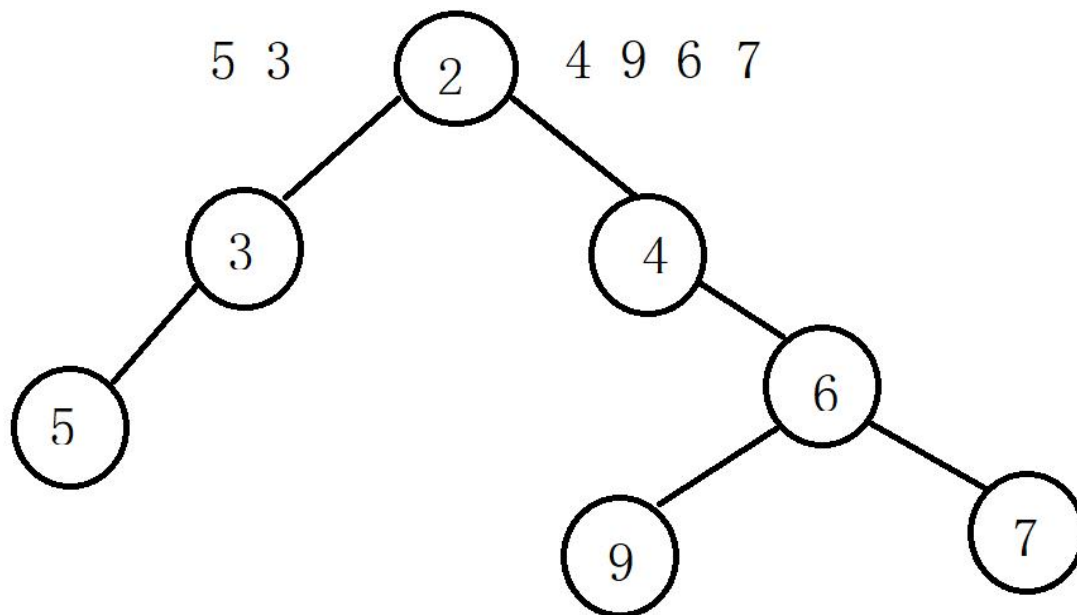
$\text{sum}(\text{depth} * b[\text{mink}])$

```
7 int f(int l, int r, int depth) {
8     if(l > r)
9         return 0;
10    int min = maxn, mink;
11    for(int i = l; i <= r; ++i) {
12        if (min > a[i]) {
13            min = a[i];
14            mink = i;
15        }
16    }
17    int lres = f(l, mink - 1, depth + 1);
18    int rres = f(mink + 1, r, depth + 1);
19    return lres + rres + depth * b[mink];
20 }
```

假设输入

$n=7$

5 3 2 4 9 6 9



●判断题.

1)如果a 数组有重复的数字，则程序运行时会发生错误。（ F ）

没有逻辑根据

2)如果b数组全为0，则输出为0。（ T ）

depth*b[mink],这里的b[mink]就全部是0

●选择题

3) 当n=100时，最坏情况下，与第12行的比较运算执行的次数最接近的是：（ A ）。

A.5000 B. 600 C.6 D.100

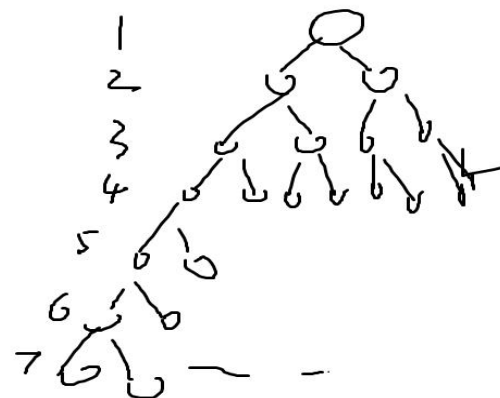
12行求最小值，最坏的情况下，输入的序列[从大到小排]，是一颗斜二叉树。因此比较100次，查找的次数为
 $100+99+98+\dots+3+2+1=5050$

```
7 int f(int l, int r, int depth) {  
8     if(l>r)  
9         return 0;  
10    int min = maxn, mink;  
11    for(int i=l; i<=r; ++i) {  
12        if (min > a[i]) {  
13            min = a[i];  
14            mink = i;  
15        }  
16    }  
17    int lres = f(l, mink - 1, depth + 1);  
18    int rres = f(mink + 1, r, depth + 1);  
19    return lres + rres + depth * b[mink];  
20 }
```

4)当n=100时，最好情况下，与第12行的比较运算执行的次数最接近的是：（ D ）。

A. 100 B. 6 C. 5000 **D. 600**

最好的情况下，是一颗完全二叉树。树的高度为7， $1*(100)+2*(50)+4*(25)+...=600$ 。
每一层找100次，找6次。也可以从时间复杂度理解。 $n\log n=700$



```
7 int f(int l, int r, int depth) {
8     if(l>r)
9         return 0;
10    int min = maxn, mink;
11    for(int i=l; i<=r; ++i) {
12        if (min > a[i]) {
13            min = a[i];
14            mink = i;
15        }
16    }
17    int lres = f(l, mink - 1, depth + 1);
18    int rres = f(mink + 1, r, depth + 1);
19    return lres + rres + depth * b[mink];
20 }
```

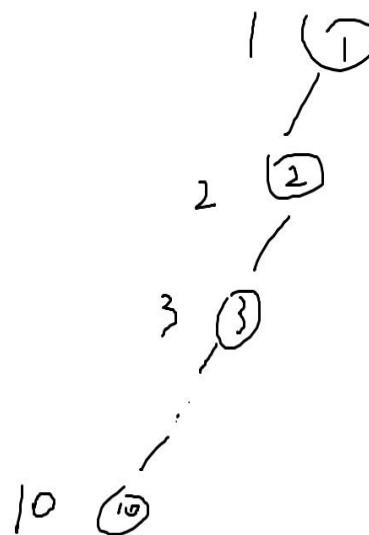
5)当n=10时，若b数组满足，对任意 $0 \leq i < n$ ，都有 $b[i]=i+1$ ，那么输出最大为（ D ）。

A.386 B.383 C.384 **D.385**

输出最大为斜二叉树，树高为1~10的点各一个，

$$\text{sum} = \sum i^2 = 1*1 + 2*2 + \dots + 10*10 = 385$$

```
7 int f(int l, int r, int depth) {  
8     if(l>r)  
9         return 0;  
10    int min = maxn, mink;  
11    for(int i=l; i<=r; ++i) {  
12        if (min > a[i]) {  
13            min = a[i];  
14            mink = i;  
15        }  
16    }  
17    int lres = f(l, mink - 1, depth + 1);  
18    int rres = f(mink + 1, r, depth + 1);  
19    return lres + rres + depth * b[mink];  
20 }
```

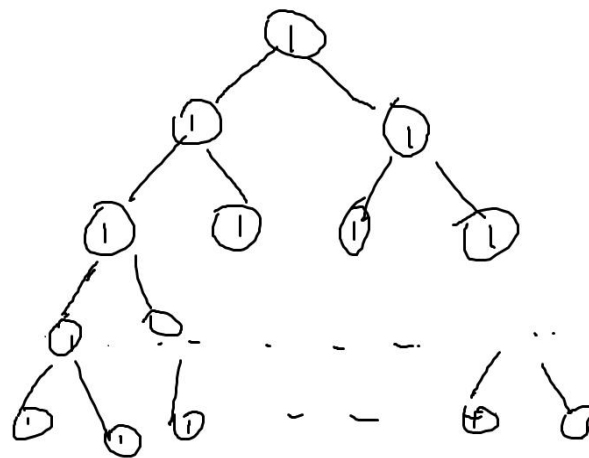


6) (4分)当 $n=100$ 时, 若 b 数组满足, 对任意 $0 \leq i < n$, 都有 $b[i]=1$, 那么输出最小为 (B)。

A. 582 **B.580** C.579 D.581

完全二叉树的时候最小

$\text{sum}=1*1+2*2+4*3+8*4+16*5+32*6+37*7=580$



```
7 int f(int l, int r, int depth) {
8     if(l>r)
9         return 0;
10    int min = maxn, mink;
11    for(int i=l; i<=r; ++i) {
12        if (min > a[i]) {
13            min = a[i];
14            mink = i;
15        }
16    }
17    int lres = f(l, mink - 1, depth + 1);
18    int rres = f(mink + 1, r, depth + 1);
19    return lres + rres + depth * b[mink];
20 }
```

4 完善程序1[6327] 矩阵变幻

1. (矩阵变幻) 有一个奇幻的矩阵，在不停的变幻，其变幻方式为：数字0变成矩阵，

0 0

0 1

数字1变成矩阵

1 1

1 0

最初该矩阵只有一个元素0，变幻n次后，矩阵会变成什么样？

例如，矩阵最初为：[0]；矩阵变幻1次后：

0 0

0 1

矩阵变幻2次后：

0 0 0 0

0 1 0 1

0 0 1 1

0 1 1 0

4 完善程序1

输入一行一个不超过10的正整数n。输出变幻n次后的矩阵。
试补全程序。

提示：

“<<”表示二进制左移运算符，例如 $(11)_2 \ll 2 = (1100)_2$ ；

而“^”表示二进制异或运算符，它将两个参与运算的数中的每个对应的二进制位一一进行比较，若两个二进制位相同，则运算结果的对应二进制位为0，反之为1。

```
3 int n;
4 const int max_size = 1 << 10;
5 int res[max_size][max_size];
6 void recursive(int x, int y, int n, int t) {
7     if(n==0) {
8         res[x][y] = 1; //1
9         return;
10    }
11    int step=1<<(n-1);
12    recursive(__2__, n - 1, t); //2
13    recursive(x, y + step, n - 1, t); //
14    recursive(x + step, y, n - 1, t); //
15    recursive(__3__, n - 1, !t); //3
16 }
17 int main() {
18     scanf("%d", &n);
19     recursive(0, 0, __4__);
20     int size = __5__; //4
21     for(int i=0; i<size; ++i) {
22         for(int j=0; j<size; ++j)
23             printf("%d", res[i][j]);
24         puts("");
25     }
26     return 0;
27 }
```

3 完善程序1[6537] 矩阵变幻

```
3 int n;
4 const int max_size = 1 << 10;
5 int res[max_size][max_size];
6 void recursive(int x, int y, int n, int t) {
7     if(n==0) {
8         res[x][y] = t;
9         return;
10    }
11    int step=1<<(n-1);
12    recursive(x, y, n - 1, t); //左上角
13    recursive(x, y + step, n - 1, t); //右上角
14    recursive(x + step, y, n - 1, t); //左下角
15    recursive(x+step, y+step, n - 1, !t); //右下角
16 }
17
```


3 完善程序

```
17
18 int main() {
19     scanf("%d", &n);
20     recursive(0, 0, n, 0);
21     int size = 1<<n; // 矩阵的宽度变化为  $2^n = 1 \ll n$ 
22     for(int i=0; i<size; ++i) {
23         for(int j=0; j<size; ++j)
24             printf("%d", res[i][j]);
25         puts("");
26     }
27     return 0;
28 }
```

3 完善程序

1)①处应填 ()

- A. $n\%2$ B. 0 C. t D. 1

2)②处应填 ()

- A. x-step,y-step B. x, y- step C. x-step,y D. x,y

3)③处应填 ()

- A. x-step,y-step B. x+step,y+step
C. x-step,y D. x,y-step

4)④处应填 ()

- A. $n-1, n\%2$ B. n,0 C. n, $n\%2$ D. $n-1,0$

5)⑤处应填 ()

- A. $1 \ll (n+1)$ B. $1 \ll n$ C. $n+1$ D. $1 \ll (n-1)$

首先确定的是

3 [6329] 完善程序2 计数排序

2.(计数排序) 计数排序是一个广泛使用的排序方法。下面的程序使用双关键字计数排序，将 n 对10000以内的整数，从小到大排序。

例如有三对整数(3,4)、(2,4)、(3,3)，那么排序之后应该是(2,4)、(3,3)、(3,4)。输入第一行为 n ,接下来 n 行，第 i 行有两个数 $a[i]$ 和 $b[i]$ ，分别表示第 i 对整数的第一关键字和第二关键字。

从小到大排序后输出。数据范围 $1 \leq n \leq 10^7, 1 \leq a[i], b[i] \leq 10^4$ 。

提示:应先对第二关键字排序，再对第一关键字排序。数组`ord[]`存储第二关键字排序的结果，数组`res[]`存储双关键字排序的结果。试补全程序。

输入

7

5 3

2 5

3 6

4 5

3 4

7 6

6 4

输出

2 5

3 4

3 6

4 5

5 3

6 4

7 6

提示:应先对第二关键字排序, 再对第一关键字排序。数组ord[]存储第二关键字排序的结果, 数组res[]存储双关键字排序的结果。

可以看做a[i], 看做十位数排(第二关键字), 在按照个位数排(第一关键字)

输入

53 25 36 45 34 76 64

7

5 3

2 5

3 6

4 5

3 4

7 6

6 4

53 34 64 25 45 36 76

提示:应先对第二关键字排序, 再对第一关键字排序。数组ord[]存储第二关键字排序的结果, 数组res[]存储双关键字排序的结果。

可以看做a[i], 看做十位数排 (第二关键字), 在按照个位数排 (第一关键字)

```
7 int n;  
8 unsigned a[maxn], b[maxn], res[maxn], ord[maxn];  
9 unsigned cnt[maxs + 1];
```

```
11 int main() {  
12     scanf("%d", &n);  
13     for(int i=0;i<n;++i)  
14         scanf("%d%d", &a[i], &b[i]);  
15     memset(cnt, 0, sizeof(cnt));  
16     for(int i=0;i<n;++i)  
17         ____1____; // 利用cnt数组统计数量  
18     for(int i=0;i<maxs;++i)  
19         cnt[i + 1] += cnt[i];  
20     for(int i=0;i<n;++i)  
21         ____2____; // 记录初步排序结果  
22     memset(cnt, 0, sizeof(cnt));  
23     for(int i=0;i<n;++i)  
24         ____3____ // 利用cnt数组统计数量  
25     for(int i=0;i<maxs;++i)  
26         cnt[i + 1] += cnt[i];  
27     for(int i=n-1;i>=0;--i)  
28         ____4____ // 记录最终排序结果  
29     for(int i=0;i<n;++i)  
30         printf("%d %d\n", ____5____);  
31     return 0;  
32 }
```

```
11 int main() {
12     scanf("%d", &n);
13     for(int i=0;i<n;++i)
14         scanf("%d%d", &a[i], &b[i]);
15     memset(cnt, 0, sizeof(cnt));
16     for(int i=0;i<n;++i)
17         ++cnt [b[i]]; //利用cnt数组统计数量
18     for(int i=0;i<maxs;++i)
19         cnt[i + 1] += cnt[i];
20     for(int i=0;i<n;++i)
21         ord[--cnt[b[i]]] = i; // 记录初步排序结果B. x, y- step
22     memset(cnt, 0, sizeof(cnt));
23     for(int i=0;i<n;++i)
24         ++cnt[a[i]]; //利用cnt数组统计数量
25     for(int i=0;i<maxs;++i)
26         cnt[i + 1] += cnt[i];
27     for(int i=n-1;i>=0;--i)
28         res[--cnt[a[ord[i]]]] = ord[i]; // 记录最终排序结果
29     for(int i=0;i<n;++i)
30         printf("%d %d\n", a[res [i]], b[res[i]]);
31     return 0;
32 }
```



```
memset(cnt, 0, sizeof(cnt));
for (int i = 0; i < n; ++i)
    ++cnt[b[i]]; // ①利用 cnt 数组统计数量，个位数桶计数
for (int i = 0; i < maxs; ++i)
    cnt[i + 1] += cnt[i]; //桶计数前缀和
for (int i = 0; i < n; ++i)
    ord[--cnt[b[i]]] = i; // ②记录初步排序结果，个位数升序排序
memset(cnt, 0, sizeof(cnt));
for (int i = 0; i < n; ++i)
    ++cnt[a[i]]; // ③利用 cnt 数组统计数量，十位数桶计数
for (int i = 0; i < maxs; ++i) //十位数桶计数前缀和
    cnt[i + 1] += cnt[i];
for (int i = n - 1; i >= 0; --i) //按照个位数降序，
    res[--cnt[a[ord[i]]]] = ord[i]; // ④记录最终排序结果
for (int i = 0; i < n; i++)
    printf("%d %d", a[res[i]], b[res[i]]); // ⑤
```



```
// 个位桶计数
for (int i = 0; i < n; ++i)
    ++cnt[b[i]];
// 桶计数前缀和
for (int i = 0; i < maxs; ++i)
    cnt[i + 1] += cnt[i];
// 个位数升序排序
for (int i = 0; i < n; ++i)
    ord[--cnt[b[i]]] = i;
```

个位数如何排序	11、22、31、12、21、32					
	[0]	[1]	[2]	[3]	[4]	[5]
b[]	1	2	1	2	1	2
桶计数	cnt[1]=3	11、31、21		个位数是1的有3个数字		
	cnt[2]=3	22、12、32		个位数是2的有3个数字		
桶计数前缀和						
	cnt[1]=3	个位升序个位是1的最大的排名是3（其他个位1的排名2、1）				
	cnt[2]=6	个位升序个位是2的最大的排名是6（其他个位2的排名5、4）				
ord[]	[0]	[1]	[2]	[3]	[4]	[5]
i=0			0			
i=1						1
i=2		2				
i=3					3	
i=4	4					
i=5					5	
	21	31	11	32	12	22

今天的课程结束啦.....



下课了...
同学们再见!