2017初赛讲解

信智学院　陈琰宏

1 单项选择题

1. 在 8 位二进制补码中，10101011 表示的数是十进制下的（ B）。

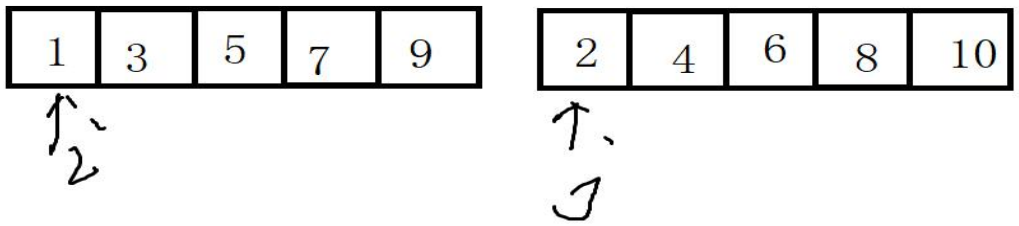
A. 43　　B. -85　　C. -43　　D. -84

2. 计算机存储数据的基本单位是（ B）。

A. bit　　B. Byte　　C. GB　　D. KB

3.设 A 和 B 是两个长为 n 的有序数组，现在需要将 A 和 B 合并成一个排好序的数组，任何以元素比较作为基本运算的归并算法在最坏情况下至少要做（ D）次比较。

A.n2　　B. n log n　　C. 2n　　D. 2n - 1



4. 分辨率为 800×600、16 位色的位图，存储图像信息所需的空间为（ A）。

A. 937.5KB　　B. 4218.75KB　　C. 4320KB　　D. 2880KB

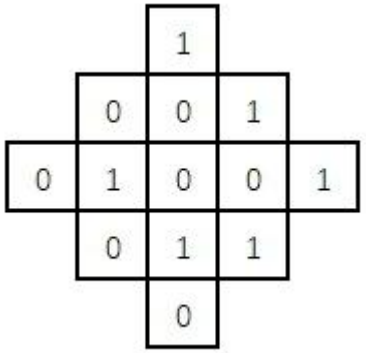
800*600*16/8/1000=

1 单项选择题

5. 一家四口人，至少两个人生日属于同一月份的概率是（C ）（假定每个人生日属于每个月份的概率相同且不同人之间相互独立）。

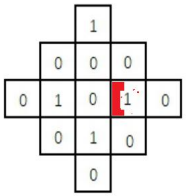
A. 1/12　　B. 1/144　　C. 41/96　　D. 3/4

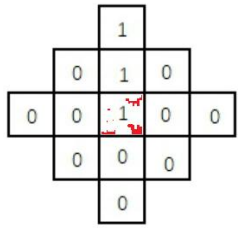
1-(全部都不在一个月的情况)=1-（12*11*10*9）/(12^4)=41/9



6. 如下图所示，共有 13 个格子。对任何一个格子进行一次操作，会使得它自己以及与它上下左右相邻的格子中的数字改变（由 1 变 0，或由 0 变 1）。现在要使得所有的格子中的数字都变为 0，至少需要（ A ）次操作。

A. 3　　B. 4　　C. 5　　D. 6





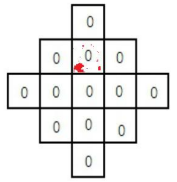
要想最少步地把所有归0，那么每次就要改变尽可能多的1。

1. 观察可得：操作第三行的第四个数，可以改变更多的1；

2. 把操作之后的图画出，继续观察，可得：操作第三行的第三个数，可以改变更多的1；

3. 把操作之后的图画出，继续观察，可得：操作第一行的第一个数，可以改变更多的1；

然后便全部归0，3步结束。



1 单项选择题

7. 对于入栈顺序为 a，b，c，d，e，f，g 的序列，下列（C ）不可能是合法的出栈序
列。

A. a，b，c，d，e，f，g　　B. a，d，c，b，e，g，f

C. a，d，b，c，g，f，e　　D. g，f，e，d，c，b，a

8. 2017 年 10 月 1 日是星期日，1999 年 10 月 1 日是（C ）。

A. 星期三　　B. 星期日　　C. 星期五　　D. 星期二

1999.10~2017.10=18，这18年里，共有 2000,2004,2008,2012,2016 共计5个闰年，所以要向后挪5天，周日开始挪，即挪到星期五。

9.甲、乙、丙三位同学选修课程，从 4 门课程中，甲选修 2 门，乙、丙各选修3门，则不同的选修方案共有（ C）种。

A. 36　　B. 48　　C. 96　　D. 192

甲乙丙彼此间独立，按步骤选择 C(4,2)*(4,3)*(4,3)=6*4*4=96

1 单项选择题

10. 设 G 是有 n 个结点、m 条边（n ≤ m）的连通图，必须删去 G 的（A ）条边，才能使得 G 变成一棵树。

A. m - n + 1　　B. m - n　　C. m + n + 1　　D. n - m + 1

n个顶点的树拥有n-1条边，所以m-(n-1)=m-n+1

11. 对于给定的序列$\{a_k\}$，我们把（i, j）称为逆序对当且仅当 i < j 且 $a_i > a_j$。那么序列 1, 7, 2, 3, 5, 4 的逆序对数为（B ）个。

A. 4　　B. 5　　C. 6　　D. 7

每个数值查看右侧右几个数比自己小。(7,2),(7,3),(7,5),(7,4),(5,4)

12. 表达式 a * (b + c) * d 的后缀形式是（ B）。

A. a b c d * + *　　B. a b c + * d *

C. a * b c + * d　　D. b + c * a * d

1 单项选择题

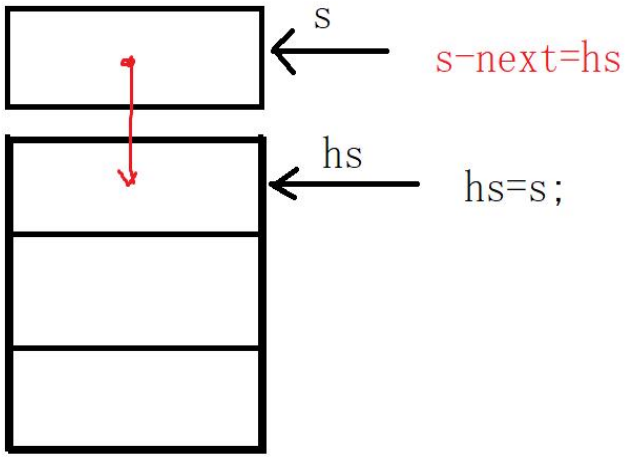
13. 向一个栈顶指针为 hs 的链式栈中插入一个指针 s 指向的结点时，应执行（ B）。

A. hs->next = s;

B. s->next = hs; hs = s;

C. s->next = hs->next; hs->next = s;

D. s->next = hs; hs = hs->next;



14. 若串 S = “copyright”，其子串的个数是（C ）。

A. 72　　B. 45　　C. 46　　D. 36

9+8+7+..+1+1=46,最后一个1为空的时候

15. 十进制小数 13.375 对应的二进制数是（ A）。

A. 1101.011　　B. 1011.011　　C. 1101.101　　D. 1010.01

2 阅读程序写结果1

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    string ch;
    int a[200];
    int b[200];
    int n, i, t, res;
    cin >> ch;
    n = ch.length();
    for (i = 0; i < 200; i++)
        b[i] = 0;
    for (i = 1; i <= n; i++) {
        a[i] = ch[i - 1] - '0';//转化为数值
        b[i] = b[i - 1] + a[i];//前缀和  位置i前面共有多少个1
    }
    res = b[n];
    t = 0;
    for (i = n; i > 0; i--) {
        if (a[i] == 0)
            t++;//在一个0、1 串中找一个位置,使得该位置前面
        if (b[i - 1] + t > res)//1的个数加上该位置后面0的个数最小,
            res = b[i - 1] + t;//最后输出最小的值。
    }
    cout << res << endl;
```

程序将读进来的字符串每一个位置的数值转化为 01 存在a数组里,而b数组的意义是a数组的前缀和,b[i]也可以理解为“位置i前面共有多少个1”,随后倒序枚举,每次遇到0就将t加一,也就是说t的意义是“位置i之后有多少个0”。

所以此程序的功能是在一个0、1 串中找一个位置,使得该位置前面1的个数加上该位置后面0的个数最小,最后输出最小的值。

2 阅读程序写结果1

判断题

(1)输入的字符串长度必须小于等于200,否则可能会出现运行时错误。(T)

(1)题目中的数组大小为200

(2)若输入字符串为10011010110011011010111110001,则会输出12 (F)11

(3)去掉第10行和第11行,程序一定可以正常运行。(F)

去掉后,b数组没有初始化,答案不能保证

(4)输入的字符串必须由01字符组成。 (F)

输入字符可以由数字组成

选择题

(5)程序的时间复杂度为(C)。

A.O(1) B. O(nlogn) C. O(n) D.O(n2)

(6)若输入10104050104010901090801OO,输出为(D 47)。

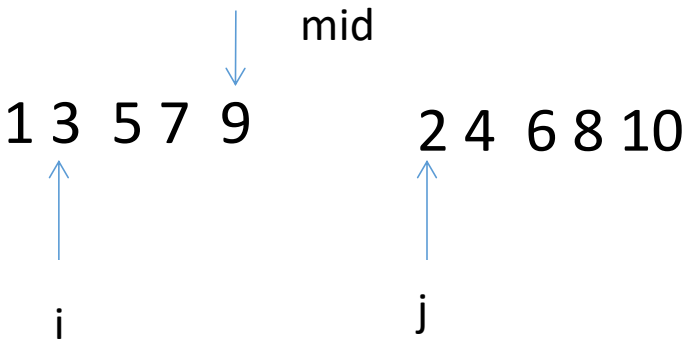
A.0 B. 5 C.26 D.47

b[i]表示前i个数字的和,于是程序的功能变成在一个数字串中找一个位置,使得该位置前面数字的和加上该位置后面0的个数最小,模拟即可

2 阅读程序写结果2

```
4  void mergesort(int l, int r){
5      if (l == r)
6          return;
7      int mid = (l + r) / 2;
8      int p = l;
9      int i = l;
10     int j = mid + 1;
11     mergesort(l, mid);
12     mergesort(mid + 1, r);
13     while (i <= mid && j <= r)
14     {//用归并排序求逆序对个数
15         if (a[j] < a[i])
16         {
17             s += mid - i + 1;//s求逆序对个数
18             t[p] = a[j];
19             p++;
20             j++;
21         }
22         else
23         {
24             t[p] = a[i];
25             p++;
26             i++;
27         }
28     }
29     while (i <= mid)
30     {
31         t[p] = a[i];
32         p++;
33         i++;
34     }
35     while (j <= r)
36     {
37         t[p] = a[j];
38         p++;
39         j++;
40     }
41     for (i = l; i <= r; i++)
42         a[i] = t[i];
43 }
44 int main( )
45 {
46     cin >> n;
47     for (i = 1; i <= n; i++)
48         cin >> a[i];
49     mergesort(1, n);
50     cout << s << endl;
51     return 0;
52 }
```

3 阅读程序写结果2



如果a[i]>a[j]，表明【a[i],a[mid]】件的元素都大于a[j],共mid-i+1个

●判断题

(1)如果将第10行的mid+1改成mid,不会影响程序结果和时间复杂度。(F)

错误，如反例 321,错误程序会输出0。

3 2 1

i

j

(2)如果将第7行的(l+R)/2改成(l+r+1)/2不会影响程序结果和时间复杂度。(T)

仍然可以保证程序会递归 O(logn)层,每个元素只会被访问O(logn)

(3)程序输出结果不可能是0。(F)

根据分析,在序列是单调不降序列时会输出0,如3\n123

●选择题

(4)该程序的时间复杂度是(B)。

A.O(n) B.O(nlogn) C.O(n √ n) D.O(n^2 /w)

(5)如果输入6\n 2 6 3 4 5 1,输出(D)。

A.1 B.2 C.4 D.8 (2,1) (6,3)(6,4)(6,5),(6,1),(3,1)(4,1)(5,1)

(6)该程序是求输入序列的(C)。

A.元素总和 B.逆序列 C.逆序对 D.卷积

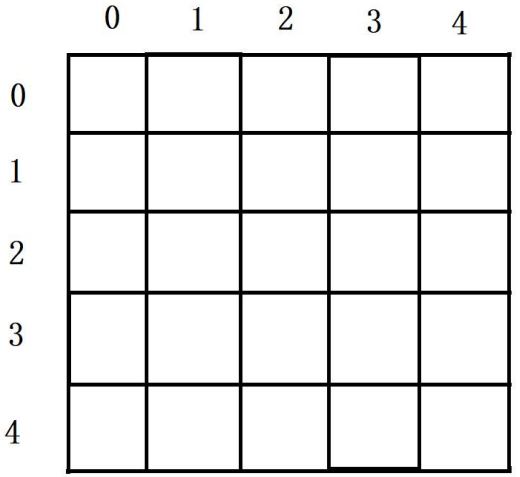
对

2 阅读程序写结果3

```
3  int main(){//一个幻方,看出来的直接模拟
4      int n,i,j,x,y,nx,ny;
5      int a[40][40];
6      for(i=0; i<40;i++)
7          for (j=0;j<40;j++)
8              a[i][j]=0;
9      cin>>n;       //n=3
10     y=0;x=n-1;   // y=0 x=2
11     n=2*n-1;     //5
12     for (i=1;i<=n*n;i++) {
13         a[y][x]=i;
14         ny= (y-1+n) %n;
15         nx= (x+1) %n;
16         if ((y==0&&x==n-1) ||a[ny][nx]!=0)
17             y=y+1;
18         else {y=ny;x=nx; }
19     }
20     for (j=0;j<n;j++)
21         cout<<a[0][j]<<" ";
22     cout<<endl;
```

第一行中间是 1,下一个数写在上一个数的右上格(第一行的上一种是最后一行，最后一列的右面是第一列),如果右上格已经填过就填它右上的下面那个格(能填右上填右上,填不了右上就填右侧那个)。

n=3



(1)11行改为n= (n<<1)-1;程序运行结果不变。(T)

n*2等价于n<<1。

(2)输出的数字有n(n是最初输入的数值)个。(F)

输出的数有(n*2-1)个。

(3)将12行的i++改为++i程序运行结果不变。(T)

for循环中i++和++i等价。

(4)将06~08行去掉，程序运行结果不变。(F)

去掉06-08行数组a定义的初始值不一定为0

选择题

(5)输入为3时，输出为(A)。

A.17 24 1 8 15　　B.17 21 3 5 2　　C.1 5 36 7 5　　D.17 2 8 3 1

(6)输入为4时，输出为(A)。

A.30 39 48 1 10 19 28　　B.24 39 48 1 10 19 28

C.30 39 48 1 5 19 14　　D.30 39 24 1 10 19 28

3 完善程序1[6580]

1.（快速幂）请完善下面的程序，该程序使用分治法求x^p mod m 的值。
输入：三个不超过 10000 的正整数 x，p，m。
输出：x^p mod m 的值。
提示：若 p 为偶数，$xp=(x^2)^{p/2}$；若 p 为奇数，$xp=x*(x^2)^{(p-1)/2}$。

```
3  int x, p, m, i, result;
4  int main() {
5      cin >> x >> p >> m;
6      result = (1)  ; //1   result 是乘法的初值,为1;
7      while ((2)  ) { //p>0 或 p!=0 或 p  如果指数还有就继续做;
8          if (p % 2 == 1)
9              result =  (3)  ; // 指数为奇数,则做一次乘法;
10         p /= 2;
11         x =  (4) ;  // 指数为偶数,则做一次倍增;
12     }
13     cout << (5) << endl;
14     return 0;
15 }
```

快速幂算法的原理是通过将指数拆分成几个因数相乘的形式，来简化幂运算。在我们计算 的时候，普通的幂运算算法需要计算3^{13}次，但是如果我们将它拆分成，再进一步拆分成 $3^8 * 3^4 * 3^1$

只需要计算4次。

这种拆分思想其实就是借鉴了二进制与十进制转换的算法思想，我们知道13的二进制是1101，可以知道：

$$13=1*2^3+1*2^2+0*2^1+1*2^0=8+4+1$$

```
int __pow(int a,int b){
    int ans = 1;
    while(b--){
        ans *= a;
    }
    return ans;
}
```

```
int __poww(int a,int b){
    int ans = 1;
    while(b){
        if(b & 1 != 0){
            ans *= a;
        }
        a *= a;
        b >>= 1;
    }
    return ans;
}
```

3 完善程序1[6537]

```
#include<bits/stdc++.h>
using namespace std;
int x, p, m, i, result;
int main() {
    cin >> x >> p >> m;
    result = 1; //1   result 是乘法的初值,为1;
    while (p) { //p>0 或 p!=0 或 p   如果指数还有就继续做;
        if (p % 2 == 1)
            result =result*x%m; // 指数为奇数,则做一次乘法;
        p /= 2;
        x = x*x%m;  // 指数为偶数,则做一次倍增;
    }
    cout << result << endl;
    return 0;
}
```

3 完善程序

(1)1处应填(B)。

A.0 B.1 C.p/2 D.p-1/2

(2)2处应填(A)。

A. p B. x C.p/2!=0 D.p/2

(3)3处应填(C)。

A. result%m B. x%m

C. result*x%m D.result*result

(4)4处应填(B)。

A. x*x B.x*x%m C.x%m D.x*x%result

(5)⑤处应填(D)。

A. x B.m C. p D.result

3 完善程序2

切割绳子）有 n 条绳子，每条绳子的长度已知且均为正整数。绳子可以以任意正整数长度切割，但不可以连接。现在要从这些绳子中切割出 m 条长度相同的绳段，求绳段的最大长度是多少。
输入：第一行是一个不超过 100 的正整数 n，第二行是 n 个不超过106的正整数，表示每条绳子的长度，第三行是一个不超过108的正整数 m。
输出：绳段的最大长度，若无法切割，输出 Failed。

```
int n, m, i, lbound, ubound, mid, count;
int len[100]; // 绳子长度
int main() {
    cin >> n;
    count = 0;
    for (i = 0; i < n; i++) {
        cin >> len[i];  //   累加总长度;
          (1)
    }
    cin >> m;
    if (  (2)   ) { //如果以单位长度切割都达不到 m段,则 Failed;
        cout << "Failed" << ndl;
        return 0;
    }
    lbound = 1;
    ubound = 1000000;
    while (   (3) ) {   //
        mid = (4) ;//
        count = 0;
        for (i = 0; i < n; i++)
            (5)  //
        if (count < m) ubound = mid - 1;
        else
            lbound = mid;
    }
    cout << lbound << endl;
    return 0;
}
```

```cpp
int n, m, i, lbound, ubound, mid, count;
int len[100]; // 绳子长度
int main() {
    cin >> n;
    count = 0;
    for (i = 0; i < n; i++) {
        cin >> len[i];  //   累加总长度;
        count += len[i];
    }
    cin >> m;
    if (count<m) { //如果以单位长度切割都达不到 m段,则 Failed;
        cout << "Failed" << endl;
        return 0;
    }
    lbound = 1;
    ubound = 1000000;
    while (lbound<ubound) {   //
        mid = (lbound+ubound+1)/2;// 分基本操作
        count = 0;
        for (i = 0; i < n; i++)
            count+=len[i]/mid; // 累计绳子以 mid 长度切割的整段数
        if (count < m) ubound = mid - 1;
        else
            lbound = mid;
    }
    cout << lbound << endl;
```

●选择题

(1)1处应填(B)。

A.空　　B.count+=len[i]

C.count++　　D.count=count* 10+len[i]

(2)2处应填(B)。

A. count= =m　　B. count<m

C.count>m　　D.count!=m

(3)3处应填(C)。

A. count=m　　B. lbound= ubound

C.lbound<ubound　　D.lbound> ubound

(4)4处应填(A)。

A. (lbound+ ubound+1)/2　　B.(lbound+ubound-1)/2

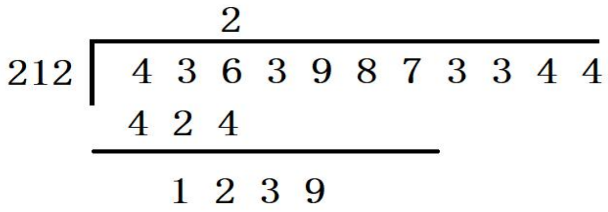
C.(ubound- lbound+1)/2　　D.(lbound+ubound+1)

(5)⑤处应填(A)。

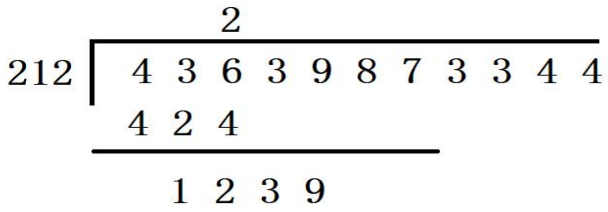
A. count+=len[i]/mid　　B.count=(count+len[i])/mid

C. count+=len[i]/lbound　　D.count=(count+len[i])/ubound

3.（大整数除法）给定两个正整数 p 和 q，其中 p 不超过 10^{100}， q 不超过 100000， 求 p 除以 q 的商和余数。
（第一空 2 分，其余 3 分） 输入：第一行是 p 的位数 n，第二行是正整数 p，第三行是正整数 q。 输出：两行，分别是 p 除以 q 的商和余数。




```
#include <iostream>
using namespace std;
int p[100];
int n, i, q, rest;
char c;
int main( )
{
    cin >> n;
    for (i = 0;i < n; i++){
        cin >> c;
        p[i] = c - '0';
    }
    cin >> q;
    rest = (1);
    i = 1;
    while ((2) && i < n){
        rest = rest * 10 + p[i];
        i++;
    }
    if (rest < q)
        cout << 0 << endl;
    else
    {
        cout << (3);
        while (i < n)
        {
            rest = (4);
            i++;
            cout << rest / q;
        }
        cout << endl;
    }
    cout << (5) << endl;
    return 0;
}
```




```
#include <iostream>
using namespace std;
int p[100];//p/q
int n, i, q, rest;
char c;
int main( )
{
    cin >> n;
    for (i = 0;i < n; i++){
        cin >> c;
        p[i] = c - '0';
    }//把字符转数字
    cin >> q;
    rest = p[0];//1利用整式计算,先取第一位p[0]
    i = 1;
    while (rest<q && i < n){//2
    //当前被除数不足以除出大于0的数，则继续添加数字
        rest = rest * 10 + p[i];
        i++;//得到被除数
    }
    if (rest < q)//表被除数小于除数
        cout << 0 << endl;
    else
    {
        cout << rest/q;//3 得到商
        while (i < n)
        {   //4 得到下一个被除数
            rest = rest%q*10+p[i] ;
            i++;
            cout << rest / q;
        }
        cout << endl;
    }
    cout <<rest%q << endl;//得到余数
    return 0;
}
```

●选择题

(1)①处应填(C　　)。

A.q[0]　　B.n　　C. p[0]　　D.0

利用整式计算,先取第一位p[0]

(2)②处应填(　A　)。

A. rest<q　　B.rest!=q　　C. rest<=q　　D.rest

当前被除数不足以除出大于0的数，则继续添加数字

(3)③处应填(　A　)。//3 得到商

A. rest/q　B. rest%q　　C. rest　　D. rest&q

(4)4处应填(　C　)。//4 得到下一个被除数

A. pow(i,10)%q+p[i]　　B. rest*10+ p[i]%q

C. rest%q*10+p[i]　　D. rest+p[i]

(5)⑤处应填(　D　)。//得到余数

A. rest-a　　B. rest　　C. restlq　　D. rest%q

今天的课程结束啦……

下课了…
同学们再见！