2016初赛讲解

信智学院　陈琰宏

1 单项选择题

1. 以下不属于无线通信技术的是（　D　）。

A. 蓝牙　　B. WiFi　　C. 北斗　　D. 以太网

2. 如果 256 种颜色用二进制编码来表示，至少需要（C　）位。

A. 6　　B. 7　　C. 8　　D. 9

3. 有 7 个一模一样的苹果，放到 3 个一样的盘子中，一共有（B ）种放法。

A. 7　　B. 8　　C. 21　　D. 37

0 0 7　0 3 4　1 3 3

0 1 6　1 1 5　2 2 3

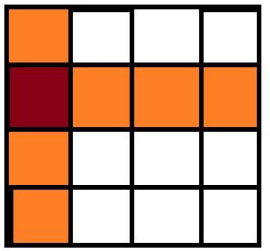
0 2 5　1 2 4

4. 从一个 4×4 的棋盘（不可旋转）中选取不在同一行也不在同一列上的两个方格，共有(　　)种方法。

A. 256　　B. 72　　C. 36　　D. 24

如图所示，16*9/2=74 ，

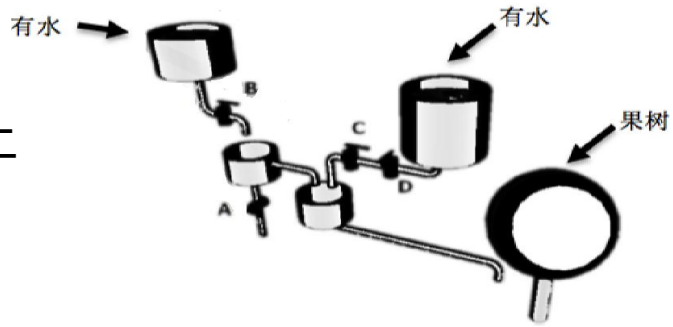
因为两个方格位置可以互换，所以重复选了



16×9= 144

1 单项选择题

5. 下图表示一个果园灌溉系统，有 A、B、C、D 四个阀门，每个阀门可以打开 或关上，所有管道粗细相同，以下设置阀门的方法中，可以让果树浇上水的果树是（A）



A. B 打开，其他都关上　　B. AB 都打开，CD 都关上
C. A 打开，其他都关上　　D. D 打开，其他都关上

6.如果开始时计算机处于小写输入状态，现在有一只小老鼠反复按照 CapsLock、 字母键 A、字母键 S 和字母键 D 的顺序循环按键，即 CapsLock、A、S、D、CapsLock、A、S、D、……，屏幕上输出的第 81 个字符是字母（ c ）。

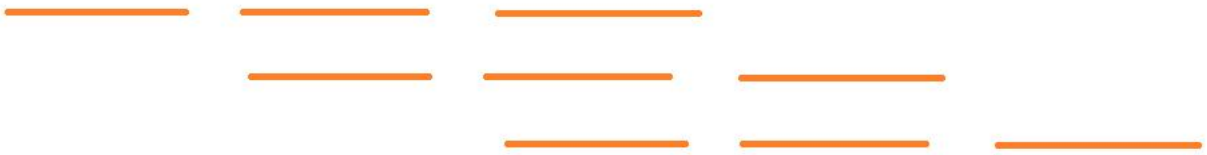
A.A　　B. S　　，C.D　　D.a

能显示的字母为A S D 三个，81%3=0，显示的必然是D(d),;81/3=27,27是奇数，所以显示的奇数的CapLock，为大写。

1 单项选择题

7.周末小明和爸爸妈妈三个人一起想动手做三道菜。小明负责洗菜、爸爸负责 切菜、妈妈负责炒菜。假设做每道菜的顺序都是：先洗菜 10 分钟，然后切 菜 10 分钟，最后炒菜 10 分钟。那么做一道菜需要 30 分钟。注意：两道不同的菜的相同步骤不可以同时进行。例如第一道菜和第二道的菜不能同时洗， 也不能同时切。那么做完三道菜的最短时间需要（C ）分钟。

A.90　　B.60　　C.50　　D.40



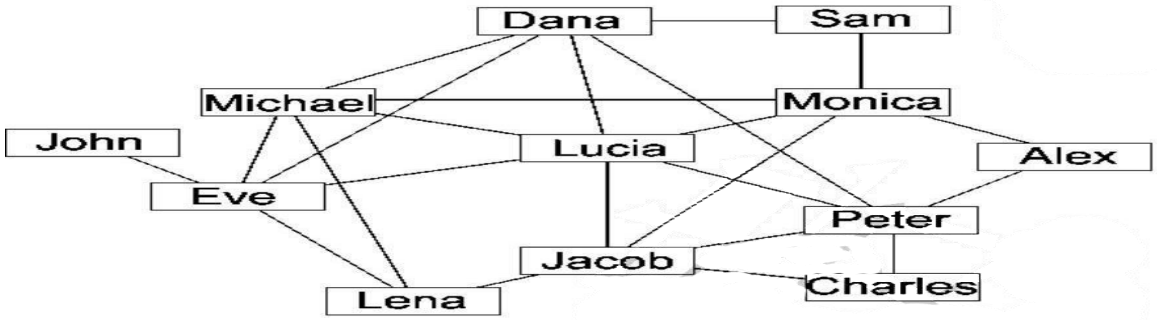
8.与二进制小数 0.1 相等的八进制数是（ B ）。

A.0.8　　B.0.4　　C.0.2　　D.0.1

1 单项选择题

9. Lucia 和她的朋友以及朋友的朋友都在某社交网站上注册了账号。下图是他 们之间的关系图，两个人之间有边相连代表这两个人是朋友，没有边相连代 表不是朋友。这个社交网站的规则是：如果某人 A 向他（她）的朋友 B 分 享了某张照片，那么 B 就可以对该照片进行评论；如果 B 评论了该照片，那 么他（她）的所有朋友都可以看见这个评论以及被评论的照片，但是不能对 该照片进行评论（除非 A 也向他（她）分享了该照片）。现在 Lucia 已经上 传了一张照片，但是她不想让 Jacob 看见这张照片，**那么她可以向以下朋友（ A ）分享该照**片。

A.Dana, Michael, Eve
B.Dana, Eve, Monica
C.Michael, Eve, Jacob
D.Micheal, Peter, Monica



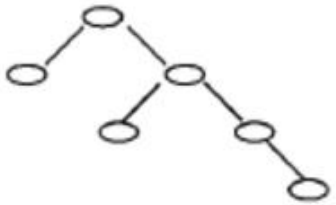
1 单项选择题

10. 以下关于字符串的判定语句中正确的是（A　　）。

A. 字符串是一种特殊的线性表　　B. 串的长度必须大于零

C. 字符串不可以用数组来表示　　D. 空格字符组成的串就是空串

11. 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），则图中所有结点的最大下标为（ D　）。



A. 6　　B. 10　　C. 12　　D. 15

12. 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;
```

则与上述程序段修改 s 值的功能等价的赋值语句是（　　B）。

A. s = a + b;　　B. s = a + c;

C. s = s + c;　　D. s = b + c;

1 单项选择题

14 给定含有 n 个不同的数的数组 L=<x 1 , x2 , ..., xn >。如果 L 中存在 xi（1 < i < n） 使得 x1 < x2 < ... < xi-1 < xi > xi+1 > ... > xn , 则称 L 是单峰的，并称 xi 是 L 的“峰顶”。现在已知 L 是单峰的，请把 a-c 三行代码补全到算法中使得算法 正确找到 L 的峰顶。

a: Search(k+1, n) b: Search(1, k-1) c:return L[k]

```
Search(1, n)
1. k<- n/2
2. if L[k] > L[k-1] and L[k] > L[k+1]
3. then _____//找到了return L[k]
4. else if L[k] > L[k-1] and L[k] < L[k+1]
5. then _____//向右找 比左边大比右边小 Search(k+1, n)
6. else _____//向左找 Search(1, k-1)
```

正确的填空顺序是（ A ）。

A. c, a, b B. c, b, a C. a, b, c D. b, a, c

1 单项选择题

13. 约定二叉树的根节点高度为 1。一棵结点数为 2016的二叉树最少有(D)个叶子结点；

A.0　　B.31　　C.32　　D.1

斜二叉树

15.设简单无向图 G 有 16 条边且每个顶点的度数都是 2，则图 G 有（D ）个顶点。

A.10　　B.12　　C.8　　D.16

度数/2=边数

顶点数*2/=16

2 阅读程序写结果1

```
#include <iostream>
using namespace std;
int main(){
    int i, length1, length2;
    string s1, s2;
    cin >> s1 >> s2;
    length1 = s1.size();
    length2 = s2.size();
    for (i = 0; i < length1; i++)
        if (s1[i] >= 'a' && s1[i] <= 'z')
            s1[i] -= 'a' - 'A';//变大写
    for (i = 0; i < length2; i++)
        if (s2[i] >= 'a' && s2[i] <= 'z')
            s2[i] -= 'a' - 'A';//变大写
    if (s1 == s2)
        cout << "=" << endl;
    else if (s1 > s2)
        cout << ">" << endl;
    else
        cout << "<" << endl;
    return 0;
}
```

程序忽略大小写的情况下比较两个字符串的大小。

2 阅读程序写结果1

(1)第6行输入的字符串如果包括数字、各类符号，可能运行错误。
（ F ） string可储存任何字符

(2)若length1 <length2.输出为“<”。 （ F ）
按字母大小比，不是比长度。

(3)若去掉第10、13行,输出结果不变。（ F ）

(4)若s1和s2中的各字符互不相同，则输出一定不为“=”。（ F ）
AB ab 相等

选择题

(5)输入Welcome与Chinese, count的结果是(C)。
A.< B.= C.> D.0
w>c

(6)若字符串的长度为n.则算法的时间复杂度是(A)。
A. O(n) B. O(nlog2n) C. $O(n^2)$ D. O(nlogn)

2 阅读程序写结果2[6614]

```
#include <iostream>
using namespace std;//程序求最长的回文子序列的长度
int lps(string seq, int i, int j){
    int len1, len2;//i为左边界，j为右边界
    if (i == j)//左右
        return 1;
    if (i > j)
        return 0;
    if (seq[i] == seq[j])//左右相等，往里走
        return lps(seq, i + 1, j - 1) + 2;//[i+1,j-1]
    len1 = lps(seq, i, j - 1);//往左[i,j-1]
    len2 = lps(seq, i + 1, j);//往右[i+1,j]
    if (len1 > len2)
        return len1;
    return len2;
}
int main(){
    string seq;cin>>seq;
    int n = seq.size();
    cout << lps(seq, 0, n - 1) << endl;
    return 0;
}
```

●判断题

(1)n代表seg的长度(T)

(2)输入acmerandacm,输出5。(T) acmerandacm,最长的回文子序列

(3)第7～8行代码删去后程序仍能正常运行。(F)

输入aa时，删除会运行错误，没有了退出条件。

(4)程序最好情况下的时间复杂度为O(n^2)。(F)

最好情况是，整个序列都是同一个字母，时间复杂度为O(n).

●选择题

(5)程序的最坏时间复杂度为(D)。

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

整个序列都是不同字母，答案区间每扩大1，复杂度就乘以2，O(2^n)

(6)函数lps(seq,i,j)用途是(C)。

A.求字符串seq的最长回文子序列长度。

B.求字符串seq中区间[i,j]上的最长回文子串长度。

C.求字符串seq中区间[i,j]上的最长回文子序列长度。

D.求字符串seq中区间[i,j]上的最长相同前缀后缀长度。

2 阅读程序3[6615]

题目中使用了邻接矩阵与 DFS,同时只读取了 n-1条边，因此可知代码是在处理一个树上问题,下一步看到统计答案部分,发现统计了最小的 weight[i]以及对应的i。

再查看 DFS函数，统计了 sum和 weight 两个量,sum 指的是子树大小(注意:以1为根情况下的子树大小,子树大小与整个树的根的位置显然是有关系的)。而 weight 是在 每个结点的最大的子树大小 和 n-sum[node] 中取最大值，其意义正是“以当前结点为根的情况下,最大的一个子树大小”。

所以,此程序求解的问题是:给定一棵树,求树的重心,使得剩下的图中,最大的连通块所包含的结点数最少。如果有多个重心,取编号最小的,输出的是删去的结点编号以及最大的连通块结点数。

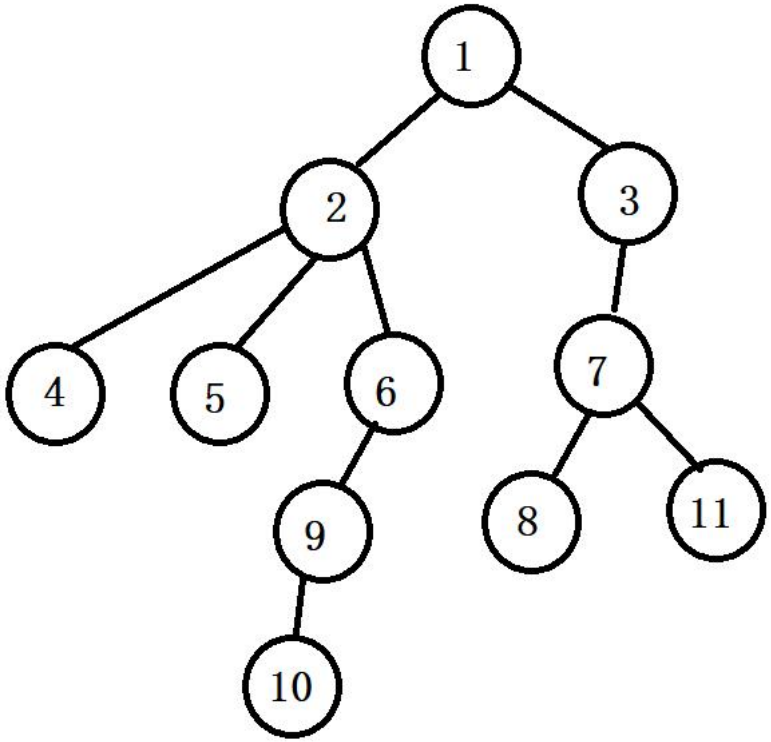
按照题意，手动模拟一下即可得到答案

找到树的重心，并输出将重心删除后，剩余各个连通块中点数的最大值。

重心定义：重心是指树中的一个结点，如果将这个点删除后，剩余各个连通块中点数的最大值最小，那么这个节点被称为树的重心。

```cpp
int map[100][100];
int sum[100], weight[100];
int visit[100];
int n;
void dfs(int node)
{
    visit[node] = 1;
    sum[node] = 1;
    int v, maxw = 0;
    for (v = 1; v <= n; v++)
    {
        if (!map[node][v] || visit[v])
            continue;
        dfs(v);
        sum[node] += sum[v];
        if (sum[v] > maxw)
            maxw = sum[v];
    }
    if (n - sum[node] > maxw)
        maxw = n - sum[node];
    weight[node] = maxw;
}
```

```cpp
int main( )
{
    memset(map, 0, sizeof(map));
    memset(sum, 0, sizeof(sum));
    memset(weight, 0, sizeof(weight));
    memset(visit, 0, sizeof(visit));
    cin >> n;
    int i, x, y;
    for (i = 1; i < n; i++)
    {//邻接矩阵 有边为1
        cin >> x >> y;
        map[x][y] = 1;
        map[y][x] = 1;
    }
    dfs(1);
    int ans = n, ansN = 0;
    for (i = 1; i <= n; i++)
        if (weight[i] < ans)
        {
            ans = weight[i];
            ansN = i;
        }
    cout << ansN << " " << ans << endl;
    return 0;
}
```


1
2
3
4
5
6
7
8
9
11
10

3 阅读程序写结果3

判断题

(1)ansN和ans 的值不可能一样。 (F)

(2)25～28行删掉后程序照样输出正常结果。 (T)

(3)本程序的时间复杂度为O(n)。 (F)

(4)38行的i++改为++i程序照样输出正常结果。 (T)

3 阅读程序写结果3

选择题

(5)输入为

1 1
1 2
1 3
2 4
2 5
2 6
3 7
7 8
7 11
6 9
9 10

输出为(　A　)。

A.2 5　B.3 4　C.4 5　D.3 5

(6)n为11时,ans的值最小为(　B　)。

A.4　B.5　C.6　D.7

完善程序1[6588]：读入整数

（读入整数）请完善下面的程序，使得程序能够读入两个 int 范围内的整数，并将这两个整数分别输出，每行一个。
输入的整数之间和前后只会出现空格或者回车。输入数据保证合法。

```cpp
int readint() {
    int num = 0;     // 存储读取到的整数
    int negative = 0;    // 负数标识
    char c; // 存储当前读取到的字符
    c = cin.get();
    while ((c < '0' || c > '9') && c != '-')
        c = cin.get() ;// (1)不是数字也不是负号重新读入
    if (c == '-') //标记负数
        negative=1;
    else
        num=c-'0'; // (2) 转化为数字
    c=cin.get();
    while( c>='0'&&c<='9'  ) { // (3) 读入数字
        num=num*10+c-'0';//转化数字（4）
        c = cin.get();
    }
    if (negative == 1)//判断是否为负数
        num=-num; // (5)    正数变负数
    return num;
}
```

```cpp
int main() {
    int a, b;
    a = readint();
    b = readint();
    cout << a << endl << b << endl;
    return 0;
}
```

(1)1处应填(D)。
A.c-'0' B.'0' C.C+ '0' D.cin.get()
(2)②处应填(B)。
A. num=0 B.num=c-'0' C. num=c-'a' D.num=c
(3)3处应填(A)。
A.c>='0'&&c<='9’, B.c>='a'&&c<='z'
C.c<'0'|Ic>'9’ D.c!='-'
(4)④处应填(B)。
A.num= num+ c- '0' B.num=num* 10＋c-'0'
C. num=num+c D.num=num* 10+c
(5)5处应填(A)。
A. num=- num B.num=num+num C.num-- D.num++

3 完善程序2[6593]

（郊游活动）有 n 名同学参加学校组织的郊游活动，已知学校给这 n 名同学 的郊游总经费为 A 元，与此同时第 i 位同学自己携带了 Mi 元。为了方便郊 游，活动地点提供 B(≥n)辆自行车供人租用，租用第 j 辆自行车的价格为 Cj 元，每位同学可以使用自己携带的钱或者学校的郊游经费，为了方便账务管 理，每位同学只能为自己租用自行车，且不会借钱给他人，他们想知道最多 有多少位同学能够租用到自行车。

本题采用二分法。对于区间[l, r]，我们取中间点 mid 并判断租用到自行 车的人数能否达到 mid。判断的过程是利用贪心算法实现的。

3 完善程序

n个人，B辆车，A总经费，M[i]第i位学生携带的钱，C[i]第i辆车租用的价格
先将租用每辆自行车的价格和学生带的钱从小到大排序。
本题主要考查分治算法和贪心算法。check 函数使用贪心算法判断是否能够让 nn个人租到自行车。贪心策略如下:让钱最多的 nn个人租最便宜的 nn 辆车，其中,钱最少的租最便宜的车,钱第二少的租第二便宜的车……钱最多的租最贵的车。

```
4  int n,B,A,M[MAXN],C[MAXN],l,r,ans,mid;
5
6  bool check(int nn){//判断nn个人租车钱是否超过了预算
7      int count = 0,i,j;
8      i =  (1)  ;//(1) i是排序后钱多的人
9      j = 1;
10     while(i <= n){
11         if(  (2)  )//如果学生自己带的钱不够，就要用经费
12             count += C[j]-M[i];//花费的经费
13         i++;
14         j++;
15     }
16     return   (3)  ; //花费的经费没超过预算就说明还有人可以租自行车
17 }
```

```
33 int main()
34 {
35     int i;
36     cin >> n >> B >> A;
37     for(i = 1; i <= n; i++)
38         cin >> M[i];
39     for(i = 1; i <= B; i++)
40         cin >> C[i];
41     sort(M,1,n);
42     sort(C,1,B);
43     l = 0;
44     r = n;
45     while(l <= r){
46         mid = (l+r)/2;
47         if(    (4)     )
48         {//如果没有超过经费，能租车的人数就比mid多
49             ans = mid;
50             l = mid+1;
51         }
52         else
53             r =     (5)     ;
54     }
55     cout << ans << endl;
56     return 0;
57 }
```

```cpp
#include<iostream>
using namespace std;
#define MAXN 1000000
int n,B,A,M[MAXN],C[MAXN],l,r,ans,mid;

bool check(int nn){//判断nn个人租车钱是否超过了预算
    int count = 0,i,j;
    i = n-nn+1;//(1) 类似田忌赛马，用钱最少得同学去租最贵的车，
    j = 1;      //如果钱够，则一定符合要求是排序后钱多的人
    while(i <= n){
        if(M[i] < C[j])//如果学生自己带的钱不够，就要用经费
            count += C[j]-M[i];//花费的经费
        i++;
        j++;
    }
    return count<=A;
    //花费的经费没超过预算就说明还有人可以租自行车
}
```

```
19 void sort(int a[],int l, int r)//从小到大排序
20 {
21     int i=l,j = r,x = a[(l+r)/2],y;
22     while(i <= j)
23     {
24         while(a[i] < x) i++;
25         while(a[j] > x) j--;
26         if(i <= j){
27             y = a[i];a[i]=a[j];a[j]=y;
28             i++;j--;
29         }
30     }
31     if(i < r) sort(a,i,r);
32     if(l < j) sort(a,l,j);
33 }
```

```
34 int main()
35 {
36     int i;
37     cin >> n >> B >> A;
38     for(i = 1; i <= n; i++)
39         cin >> M[i];
40     for(i = 1; i <= B; i++)
41         cin >> C[i];
42     sort(M,1,n);
43     sort(C,1,B);
44     l = 0;
45     r = n;
46     while(l <= r){
47         mid = (l+r)/2;
48         if(check(mid))
49         {//如果没有超过经费，能租车的人数就比mid多
50             ans = mid;
51             l = mid+1;
52         }
53         else
54             r = mid -1;
55     }
56     cout << ans << endl;
57     return 0;
58 }
```

(1)1处应填(A)。

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

让最有钱的同学去租最便宜车就可以减少学校 A的消耗j=1说明车是从便宜的开始，函数入口形式参数 nn要让最有钱的 nn个同学去租

(2)2处应填(C)。

A. M[j]>C[i] B.M[i]>C[j] C.M[i]<C[j] D.M[j]<C[i]

count 是学校的支出;i++说明当前i这位同学已经租下了第j辆车,即将尝试下一个同学,如果自己带的钱够租，那就不要让学校出钱,如果自己带的钱不够,那就让学校负担 C[j]-M[i]的钱

(3)3处应填(A)。

A. count<=A B.count>=A C. count D.A-count

关系运算,返回的应该是 A够不够

(4)4处应填(C)。测试当租车的人为 mid的时候钱是否够。

A. check (mid-1) B.!check(mid) C.check (mid) D.check(mid+1)

(5)⑤处应填(D)。二分套路,租车的人太多,钱不够,要试试人更少的情况

A.break B. r=mid+1 C.r=mid D. r=mid-1

今天的课程结束啦……

下课了…
同学们再见！