

## 十月下旬腾讯，网易游戏，百度最新校园招聘笔试题集锦(第 271-330 题)

### 引言

此文[十月百度](#)，[阿里巴巴](#)，[迅雷搜狗最新面试十一题](#)已经整理了最新的面试题 70 道，本文依次整理腾讯，网易游戏，百度等各大公司最新校园招聘的笔试题，后续将继续整理十月下旬的笔/面试题。

### 腾讯 2011.10.15 校园招聘会笔试题

1、下面的排序算法中，初始数据集的排列顺序对算法的性能无影响的是（B）

A、插入排序                      B、堆排序                      C、冒泡排序                      D、快速排序

2、以下关于 Cache 的叙述中，正确的是（B）

A、CPU 中的 Cache 容量应大于 CPU 之外的 Cache 容量

B、Cache 的设计思想是在合理成本下提高命中率

C、Cache 的设计目标是容量尽可能与主存容量相等

D、在容量确定的情况下，替换算法的时间复杂度是影响 Cache 命中率的关键因素

3、数据存储于磁盘上的排列方式会影响 I/O 服务的性能，一个圆环的磁道上有 10 个物理块，10 个数据记录 R1-----R10 存放在这个磁道上，记录的安排顺序如下表所示：

物理块	1	2	3	4	5	6	7	8	9	10
逻辑记录	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10

假设磁盘的旋转速度为 20ms/周，磁盘当前处在 R1 的开头处，若系统顺序扫描后将数据放入单缓冲区内，处理数据的时间为 4ms（然后再读取下个记录），则处理这 10 个记录的最长时间为（C）

A、180ms                      B、200ms                      C、204ms                      D、220ms

4、随着 IP 网络的发展，为了节省可分配的注册 IP 地址，有一些地址被拿出来用于私有 IP 地址，以下不属于私有 IP 地址范围的是（C）（私网 IP 地址：10.0.0.0- 10.255.255.255 ； 172.16.0.0 - 172.31.255.255； 192.168.0.0-192.168.255.255。故选 C）

A、10.6.207.84                      B、172.23.30.28                      C、172.32.50.80                      D、192.168.1.100

5、下列关于一个类的静态成员的描述中，不正确的是（D）

A、该类的对象共享其静态成员变量的值                      B、静态成员变量可被该类的所有方法访问

C、该类的静态方法只能访问该类的静态成员变量

D、该类的静态数据成员变量的值不可修改

6、已知一个线性表（38，25，74，63，52，48），假定采用散列函数  $h(\text{key}) = \text{key} \% 7$  计算散列地址，并散列存储在散列表 A【0...6】中，若采用线性探测方法解决冲突，则在该散列表上进行等概率成功查找的平均查找长度为（C）

A、1.5

B、1.7

C、2.0

D、2.3

依次进行取模运算求出哈希地址：

A	0	1	2	3	4	5	6
记录	63	48		38	25	74	52
查找次数	1	3		1	1	2	4

74 应该放在下标为 4 的位置，由于 25 已经放在这个地方，所以 74 往后移动，放在了下标为 5 的位置上了。

由于是等概率查找，所以结果为： $1/6 * (1+3+1+1+2+4) = 2.0$

7、表达式“ $X=A+B*(C--D)/E$ ”的后缀表示形式可以为（C）

A、 $XAB+CDE/*=-$

B、 $XA+BC-DE/*=-$

C、 $XABCD-*E/+=-$

D、 $XABCDE+*/=-$

8、（B）设计模式将抽象部分与它的实现部分相分离。

A、Singleton（单例）

B、Bridge（桥接）

C、Composite（组合）

D、Facade（外观）

9、下面程序的输出结果为多少？

```
1. void Func(char str_arg[100])
2. {
3.     printf("%d\n", sizeof(str_arg));
4. }
5.
6. int main(void)
7. {
8.     char str[]="Hello";
9.     printf("%d\n", sizeof(str));
10.    printf("%d\n", strlen(str));
11.    char *p = str;
12.    printf("%d\n", sizeof(p));
13.    Func(str);
14. }
```

输出结果为：6 5 4 4

对字符串进行 `sizeof` 操作的时候，会把字符串的结束符“\0”计算进去的，进行 `strlen` 操作求字符串的长度的时候，不计算 \0 的。

数组作为函数参数传递的时候，已经退化为指针了，Func 函数的参数 `str_arg` 只是表示一个指针，那个 100 不起任何作用的。

10、下面程序的输出结果为多少？

```
1. void Func(char str_arg[2])
2. {
3.     int m = sizeof(str_arg);    //指针的大小为 4
4.     int n = strlen(str_arg);    //对数组求长度，str_arg 后面的那个 2 没有任何意义，数组已经退化为指针了
5.     printf("%d\n",m);
6.     printf("%d\n",n);
7. }
8. int main(void)
9. {
10.     char str[]="Hello";
11.     Func(str);
12. }
```

输出结果为：4 5

`strlen` 只是对传递给 Func 函数的那个字符串求长度，跟 `str_arg` 中的那个 2 是没有任何关系的，即使把 2 改为 200 也是不影响输出结果的。。

11、到商店里买 200 的商品返还 100 优惠券（可以在本商店代替现金）。请问实际上折扣是多少？

算法编程题：

1、给定一个字符串，求出其最长的重复子串。

思路：使用后缀数组，对一个字符串生成相应的后缀数组后，然后再排序，排完序依次检测相邻的两个字符串的开头公共部分。

这样的时间复杂度为：

生成后缀数组  $O(N)$

排序  $O(N\log N)$  最后面的  $N$  是因为字符串比较也是  $O(N)$

依次检测相邻的两个字符串  $O(N * N)$

总的时间复杂度是  $O(N^2 \log N)$ ,

网易游戏 2011.10.15 校园招聘会笔试题

1、对于一个内存地址是 32 位、内存页是 8KB 的系统。0X0005F123 这个地址的页号与页内偏移分别是多少。

2、如果 X 大于 0 并小于 65536，用移位法计算 X 乘以 255 的值为：-X+X<<8

X<<8-X 是不对的，X<<8，已经把 X 的值改变了（订正：X<<8 是个临时变量，不会改变 X 的值，就像 a+1 不会改变 a 一样）。

3、一个包含 n 个节点的四叉树，每个节点都有四个指向孩子节点的指针，这 4n 个指针中有 3n+1 个空指针。

4、以下两个语句的区别是：

```
1. int *p1 = new int[10];
2. int *p2 = new int[10]();
```

5、计算机在内存中存储数据时使用了大、小端模式，请分别写出 A=0X123456 在不同情况下的首字节是，大端模式：0X12 小端模式：0X56 X86 结构的计算机使用 小端 模式。

一般来说，大部分用户的操作系统（如 windows, FreeBSD, Linux）是小端模式的。少部分，如 MAC OS，是大端模式的。

6、在游戏设计中，经常会根据不同的游戏状态调用不同的函数，我们可以通过函数指针来实现这一功能，请声明一个参数为 int \*，返回值为 int 的函数指针：

int (\*fun)(int \*)

7、在一冒险游戏里，你见到一个宝箱，身上有 N 把钥匙，其中一把可以打开宝箱，假如没有任何提示，随机尝试，问：

（1）恰好第 K 次（1≤K≤N）打开宝箱的概率是多少。

（2）平均需要尝试多少次。

## 百度 2011.10.16 校园招聘会笔试题

### 一、算法设计

1、设 rand (s, t) 返回[s,t]之间的随机小数，利用该函数在一个半径为 R 的圆内找随机 n 个点，并给出时间复杂度分析。

2、为分析用户行为，系统常需存储用户的一些 query，但因 query 非常多，故系统不能全存，设系统每天只存 m 个 query，现设计一个算法，对用户请求的 query 进行随机选择 m 个，请给一个方案，使得每个 query 被抽中的概率相等，并分析之，注意：不到最后一刻，并不知用户的总请求量。

### 3、C++ STL 中 vector 的相关问题：

- (1)、调用 `push_back` 时，其内部的内存分配是如何进行的？
- (2)、调用 `clear` 时，内部是如何具体实现的？若想将其内存释放，该如何操作？

### 二、系统设计

正常用户端每分钟最多发一个请求至服务端，服务端需做一个异常客户端行为的过滤系统，设服务器在某一时刻收到客户端 A 的一个请求，则 1 分钟内的客户端任何其它请求都需要被过滤，现知每一客户端都有一个 IPv6 地址可作为其 ID，客户端个数太多，以至于无法全部放到单台服务器的内存 hash 表中，现需简单设计一个系统，使用支持高效的过滤，可使用多台机器，但要求使用的机器越少越好，请将关键的设计和思想用图表和代码表现出来。

### 三、求一个全排列函数：

如 `p([1,2,3])` 输出：

[123]、[132]、[213]、[231]、[321]、[323]

求一个组合函数

如 `p([1,2,3])` 输出：

[1]、[2]、[3]、[1,2]、[2,3]、[1,3]、[1,2,3]

这两问可以用伪代码。

## 迅雷 2011.10.21 笔试题

1、下面的程序可以从 1....n 中随机输出 m 个不重复的数。请填空

```
knuth(int n, int m)
```

```
{  
  
    srand((unsigned int)time(0));  
  
    for (int i=0; i<n; i++)  
    {  
        if (_____)  
        {  
            cout<<i<<endl;  
            _____;  
        }  
    }  
}
```

```
}
```

分别为: `rand()%(n-i)<m` 和 `m--`;

2、以下 `prim` 函数的功能是分解质因数。请填空

```
void prim(int m, int n)

{

    if (m>n)

    {

        while (_____) n++;

        _____;

        prim(m,n);

        cout<<n<<endl;

    }

}
```

分别为: `m%n` 和 `m/=n`

3、下面程序的功能是输出数组的全排列。请填空

```
void perm(int list[], int k, int m)

{

    if (_____)

    {

        copy(list,list+m,ostream_iterator<int>(cout," "));

        cout<<endl;

        return;

    }

}
```

```

for (int i=k; i<=m; i++)
{
    swap(&list[k],&list[i]);

    _____;

    swap(&list[k],&list[i]);
}
}

```

分别为:  $k==m$  和 `perm (list,k+1,m)`

## 二、主观题:

1、（40 分）用户启动迅雷时，服务器会以 `uid,login_time,logout_time` 的形式记录用户的在线时间；用户在使用迅雷下载时，服务器会以 `taskid,start_time,finish_time` 的形式记录任务的开始时间和结束时间。有效下载时间是指用户在开始时间和结束时间之间的在线时间，由于用户可能在下载的时候退出迅雷，因此有效下载时间并非 `finish_time` 和 `start_time` 之差。假设登录记录保存在 `login.txt` 中，每一行代表用户的上下线记录；下载记录保存在 `task.txt` 中，每一行代表一个任务记录，记录的字段之间以空格分开。计算每个用户的有效下载时间和总在线时间的比例。注意：请尽量使用 STL 的数据结构和算法

2、（60 分）在  $8 \times 8$  的棋盘上分布着  $n$  个骑士，他们想约在某一个格中聚会。骑士每天可以像国际象棋中的马那样移动一次，可以从中间像 8 个方向移动（当然不能走出棋盘），请计算  $n$  个骑士的最早聚会地点和要走多少天。要求尽早聚会，且  $n$  个人走的总步数最少，先到聚会地点的骑士可以不再移动等待其他的骑士。

从键盘输入  $n$  ( $0 < n \leq 64$ )，然后一次输入  $n$  个骑士的初始位置  $x_i, y_i$  ( $0 \leq x_i, y_i \leq 7$ )。屏幕输出以空格分隔的三个数，分别为聚会点 ( $x, y$ ) 以及走的天数。

## 盛大游戏 2011.10.22 校园招聘会笔试题

### 1、下列代码的输出为:

```

1. #include "iostream"
2. #include "vector"
3. using namespace std;
4.
5. int main(void)
6. {
7.     vector<int>array;
8.     array.push_back(100);
9.     array.push_back(300);

```

```

10.     array.push_back(300);
11.     array.push_back(500);
12.     vector<int>::iterator itor;
13.     for(itor=array.begin();itor!=array.end();itor++)
14.     {
15.         if(*itor==300)
16.         {
17.             itor = array.erase(itor);
18.         }
19.     }
20.     for(itor=array.begin();itor!=array.end();itor++)
21.     {
22.         cout<<*itor<<" ";
23.     }
24.     return 0;
25. }

```

A、100 300 300 500    B、100 300 500    C、100 500    D、程序错误

vector 在 erase 之后，指向下一个元素的位置，其实进行 erase 操作时将后面所有元素都向前移动，迭代器位置没有移动。itor=array.erase(itor) erase 返回下一个元素的地址，相当于给 itor 一个新值。

2、下列代码的输出为：

```

1. class CParent
2. {
3. public:
4.     virtual void Intro()
5.     {
6.         printf("I'm a Parent, ");
7.         Hobby();
8.     }
9.     virtual void Hobby()
10.    {
11.        printf("I like football!");
12.    }
13. };
14. class CChild:public CParent
15. {
16. public:
17.     virtual void Intro()
18.     {
19.         printf("I'm a Child, ");
20.         Hobby();
21.     }
22.     virtual void Hobby()
23.     {
24.         printf("I like basketball!\n");
25.     }
26. };

```



```

27. int main(void)
28. {
29.     CChild *pChild = new CChild();
30.     CParent *pParent = (CParent*)pChild;
31.     pParent->Intro();
32.     return 0;
33. }

```

- A、I'm a Child,I like football!                      B、I'm a Child,I like basketball!  
 C、I'm a Parent,I like football!                      D、I'm a Parent,I like basketball!

3、在 win32 平台下，以下哪种方式无法实现进程同步？

- A、Critical Section    B、Event    C、Mutex    D、Semaphore

4、以下哪句的说法是正确的

- A、在页式存储管理中，用户应将自己的程序划分为若干个相等的页  
 B、所有的进程都挂起时，系统将陷入死锁  
 C、执行系统调用可以被中断  
 D、进程优先数是进程调度的重要依据，必须根据进程运行情况动态改变

5、以下描述正确的是

- A、虚函数是可以内联的，可以减少函数调用的开销提高效率  
 B、类里面可以同时存在函数名和参数都一样的虚函数和静态函数  
 C、父类的析构函数是非虚的，但是子类的析构函数是虚的，delete 子类对象指针会调用父类的析构函数  
 D、以上都不对

简答题：快速排序的思想是递归的，但是它的平均效率却是众多排序算法中最快的，为什么？请结合本例说明你对递归程序的理解。

算法题：用你熟悉的编程语言，设计如下功能的函数：输入一个字符串，输出该字符串中所有字母的全排列。程序请适当添加注释。

C++函数原型： void Print (const char \*str)

输入样例： abc

输出结果： abc、acb、bca、bac、cab、cba

（以上部分整理自此君博客：<http://blog.csdn.net/Hackbuteer1>。十分感谢。有何不妥之处，还望海涵海涵。）

## 后续整理

- 12 个工厂分布在一条东西向高速公路的两侧，工厂距离公路最西端的距离分别是 0、4、5、10、12、18、27、30、31、38、39、47。在这 12 个工厂中选取 3 个原料供应厂，使得剩余工厂到最近的原料供应厂距离之和最短，问应该选哪三个厂？

## 7、下面程序运行后的结果为：to test something

```
01. char str[] = "glad to test something";  
02. char *p = str;  
03. p++;  
04. int *p1 = static_cast<int *>(p);  
05. p1++;  
06. p = static_cast<char *>(p1);  
07. printf("result is %s\n",p);
```

2.

3. hash 冲突时候的解决方法？

- 1)、开放地址法
- 2)、再哈希法
- 3)、链地址法
- 4)、建立一个公共溢出区

4. int main()

```
{  
    if()  
    {  
        printf("Hello ");  
    }  
    else  
    {  
        printf("World !!!");  
    }  
    return 0;  
}
```

在 if 里面请写入语句 使得打印出 hello world。

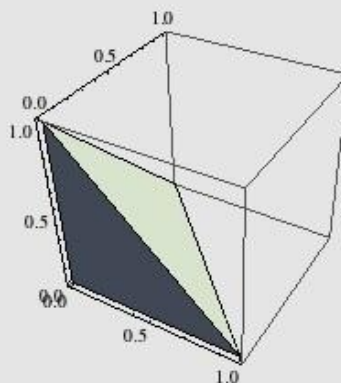
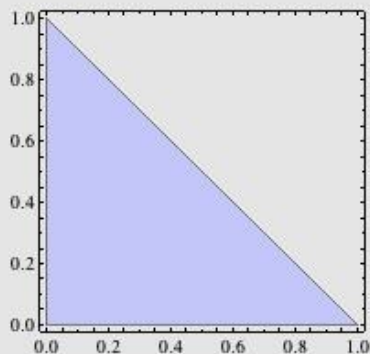
5. 今天 10.19 西山居笔试题：

分别写一个宏和函数来获取元素个数 如 count(a) 会得到 a 数组元素个数。

6. 平均要取多少个(0,1)中的随机数才能让和超过 1。（答案： e 次， 其中 e 是自然对数的底数）

为了证明这一点，让我们先来看一个更简单的问题：任取两个 0 到 1 之间的实数，它们的和小于 1 的概率有多大？容易想到，满足  $x+y < 1$  的点  $(x, y)$  占据了正方形  $(0, 1) \times (0, 1)$  的一半面积，因此这两个实数之和小于 1 的概率就是  $1/2$ 。类似地，三个数之和小于 1 的概率则是  $1/6$ ，它是平面  $x+y+z=1$  在单位立方体中截得的一个三棱锥。这个  $1/6$  可以利用截面与底面的相似比关系，通过简单的积分求得：

$$\int_{(0..1)} (x^2) * 1/2 \, dx = 1/6$$



可以想到，四个 0 到 1 之间的随机数之和小于 1 的概率就等于四维立方体一角的“体积”，它的“底面”是一个体积为  $1/6$  的三维体，在第四维上对其进行积分便可得到其“体积”

$$\int_{(0..1)} (x^3) * 1/6 \, dx = 1/24$$

依此类推，n 个随机数之和不超过 1 的概率就是  $1/n!$ ，反过来 n 个数之和大于 1 的概率就是  $1 - 1/n!$ ，因此加到第 n 个数才刚好超过 1 的概率就是

$$(1 - 1/n!) - (1 - 1/(n-1)!) = (n-1)/n!$$

因此，要想让和超过 1，需要累加的期望次数为

$$\sum_{n=2..∞} n * (n-1)/n! = \sum_{n=1..∞} n/n! = e$$

7. 今天支付宝 10.20 笔试题：汉诺塔一共为  $2*N$ ，2 个一样大小，有编号顺序 每次只能移动一个 大的不能叠在小得上面 移动完之后，相同大小的编号必须和原来一样 问最小要移动多少次？ 如 A1 A2 B1 B2 C1 C2 ..... 这样叠， $A < B < C \dots$  B 不能放 A 上面，C 不能放 B A 上面，移动到另外一个柱子后，还必须是 A1 A2 B1 B2 C1 C2 ....

8. socket 编程的问题

TCP 连接建立后，调用 send 5 次，每次发 100 字节，问 recv 最少要几次，最多要几次？

9. 迅雷笔试题：

下面的程序可以从  $1 \dots n$  中随机输出 m 个不重复的数。请填空

```
knuth(int n, int m)
{
    srand((unsigned int)time(0));
    for (int i=0; i<n; i++)
        if ( )
        {
            cout<<i<<endl;
            ( );
        }
    }
```

```
}  
}
```

10. 四个线程 t1,t2,t3,t4,向 4 个文件中写入数据, t1 只能写入 1, t2 只能写入 2, t3 只能写入 3, t4 只能写入 4, 对 4 个文件 A, B, C, D 写入如下内容

A:123412341234.....

B:234123412341....

C:341234123412....

D:412341234123....

怎么实现同步可以让线程并行工作?

11. 比如一个数组[1,2,3,4,6,8,9,4,8,11,18,19,100]

前半部分是一个递增数组, 后面一个还是递增数组, 但整个数组不是递增数组, 那么怎么最快的找出其中一个数?

12. 今日 10.21 迅雷笔试题:

1、一棵二叉树节点的定义(和平时我们定义的一样的) 它给出了一棵二叉树的根节点 说现在怀疑这棵二叉树有问题 其中可能存在某些节点不只有一个父亲节点 现要你编写一个函数判断给定的二叉树是否存在这样的节点 存在则打印出其父亲节点返回 true 否则返回 false

打印节点形式:

[当前节点][父亲节点 1][父亲节点的父亲节点][。。。]

[当前节点][父亲节点 2][父亲节点的父亲节点][。。。]

2、有一亿个整数, 请找出最大的 1000 个, 要求时间越短越好, 空间占用越少越好

13. 在频繁使用小内存时, 通常会先申请一块大的内存, 每次使用小内存时都从大内存里取, 最后大内存使用完后一次性释放, 用算法实现。

14. 今天亚马逊 A 卷校招笔试题:

输入一个字符串, 如何求最大重复出现的字符串呢? 比如输入 ttabcftgrgabcd, 输出结果为 abc, canffcancd, 输出结果为 can。

15. 今天 10.22 盛大: 删除模式串中出现的字符, 如“welcome to asted”, 模式串为“aeiou”那么得到的字符串为“wlcm t std”, 要求性能最优。

16. 数组中的数分为两组, 让给出一个算法, 使得两个组的和的差的绝对值最小

数组中的数的取值范围是  $0 < x < 100$ , 元素个数也是大于 0, 小于 100

比如  $a[] = \{2, 4, 5, 6, 7\}$ , 得出的两组数  $\{2, 4, 6\}$  和  $\{5, 7\}$ ,  $\text{abs}(\text{sum}(a1) - \text{sum}(a2)) = 0$ ;

比如  $\{2, 5, 6, 10\}$ ,  $\text{abs}(\text{sum}(2, 10) - \text{sum}(5, 6)) = 1$ , 所以得出的两组数分别为  $\{2, 10\}$  和  $\{5, 6\}$ 。

17. 百度北京研发一道系统设计题, 如何快速访问 ipv6 地址呢? ipv6 地址如何存放?

18. 百度 2012 校招北京站笔试题系统设计: 正常用户端每分钟最多发一个请求至服务端, 服务端需做一个异常客户端行为的过滤系统, 设服务器在某一时刻收到客户端 A 的一个请求, 则 1 分钟内的客户端任何其它请求都需要被过滤, 现知每一客户端都有一个 IPv6 地址可作为其 ID, 客户端个数太多, 以至于无法全部放到单台服务器的内存 hash 表中, 现需简单设计一个系统, 使用支持高效的过滤, 可使用多台机器, 但要求使用的机器越少越好, 请将关键的设计和思想用图表和代码表现出来。

19. #include <iostream>

using namespace std;

class A

{

public:

A(){cout<<"A"<<endl;}

~A(){cout<<"~A"<<endl;}

};

class B

{

public:

B(A &a):\_a(a)

{

cout<<"B"<<endl;

}

~B(){cout<<"~B"<<endl;}

private:

A \_a;

};

int main()

{

A a;

B b(a);

return 0;

// 构造次序和析构次序是对称的,这种题解答都是有技巧的.

// 拷贝构造就不说了,构造过程是:

// A A B ,那么析构必然是对称的: B A A。



```
A
B
~B
~A
~A
Press any key to continue
```

}

....

ok, 以上所有任何参考答案若有问题, 欢迎不吝指正。谢谢。日后, 继续整理十月下旬的各大 IT 公司的笔/面试题, 持续更新, 直到十月月底。祝所有诸君找到自己合适而满意的 offer, 工作。July、2011.10.17。