

探秘2016校招笔试面试



七月算法 邹博

2015年9月24日



屏幕剪辑的捕获时间: 2018/5/2 20:26

拥抱变化

□ 总体体会:

■ 面试/笔试题目越来越难了

□ 除了考察编程语言、数据结构、数据库、操作系统、计算机网络，大量增加了**算法、逻辑、数学**方面的题目，数学问题中尤其要重视**概率**；**机器学习**的题目、难度有所增加。

■ 问：为什么面试的题目充斥大量的**算法/逻辑**题？在以后的工作中要用到么？



屏幕剪辑的捕获时间: 2018/5/2 20:26

笔试题概览

移动 | 联通 00:21

阿里2016校招笔试题(算法类) 7/10

排列组合

在如下8*6的矩阵中，请计算从A移动到B一共有（ ）种走法。要求每次只能向上或向右移动一格，并且不能经过P。

							B
					P		
A							

A 456

B 492

C 568

D 626

E 680

移动 | 联通 00:05

阿里2016校招笔试题(算法类) 1/10

概率

从1,2,3,...,99,2015里任意选择一部分数（可能为0个数），这部分数按位异或起来的期望值是（ ）

A 512

B 1007

C 1008

D 2015/2

E 1024

F 2047/2

移动 | 联通 00:26

阿里2016校招笔试题(算法类) 8/10

逻辑

A、B、C、D四人应聘一个程序员职位，此职务的要求条件是：Java熟练；懂数据库开发；会web开发；有C++经验。谁满足的条件最多，谁就被雇用。

(1) 把上面四个要求条件两两组合，每个组合都恰有一人满足。同时已知

(2) A和B Java熟练

(3) B和C会web

(4) C和D懂数据库

(5) D有C++经验

那么，被雇用的是（ ）

A A

B B

C C

D D

屏幕剪辑的捕获时间: 2018/5/2 20:27

笔试题概览

移动 联通	22:27	移动 联通	22:28	移动 联通	22:26
< 00:13	< 00:48	< 00:29			
阿里2016校招笔试题(算法类) 4/10	阿里2016校招笔试题(算法类) 10/10	阿里2016校招笔试题(算法类) 7/10			
数学计算 将1,2,3,...,99,100任意排列成一个圈,相邻两数的差的绝对值求和最多为() A 100 B 198 C 200 D 500 E 2500 F 5000	概率 2015年年初,某公司部门举行年会,期间有个游戏,规则是通过猜拳的方法决出每一局的胜负(一人赢其他所有人才算赢得本局,出现平局就继续猜拳),如果谁先赢满7局,则获胜,并赢得全部奖金,现在有3名员工甲、乙、丙参与了这个游戏,游戏进行了一会,分数为甲6,乙5,丙4,但时间已晚,大家要吃饭了,最公平的奖金分配方案是() A 甲得到19/27,乙得到7/27,丙得到1/27 B 甲得到18/27,乙得到8/27,丙得到1/27 C 甲得到17/27,乙得到7/27,丙得到1/27 D 甲得到19/27,乙得到6/27,丙得到1/27	概率 商品推荐场景中过于聚焦的商品推荐往往会损害用户的购物体验,在有些场景中,系统会通过一定程度的随机性给用户带来发现的惊喜感。假设在某推荐场景中,经计算A和B两个商品与当前访问用户的匹配度分别为0.8分和0.2分,系统将随机为A生成一个均匀分布于0到0.8的最终得分,为B生成一个均匀分布于0到0.2的最终得分,那么最终B的分数大于A的分数的概率为() A 1/16 B 1/8 C 3/16 D 3/8 E 1/4			

屏幕剪辑的捕获时间: 2018/5/2 20:27

笔试题概览

移动 联通	23:38	移动 联通	23:40	移动 联通	23:42
< 00:10	< 01:55	< 00:11			
阿里2016校招笔试题(算法类) 2/10	阿里2016校招笔试题(算法类) 6/10	新浪2016校招笔试题 10/10			
计算机基础 如果下列的公式成立: $84 \times 148 = B6A8$ 。则采用的是()进制表示的。 A 15 B 11 C 12 D 14 E 16 F 以上都不对	概率 袋子中分别一叠纸币,其中5元面值的纸币6张,10元面值的纸币5张,20元面值的纸币4张,从袋子中任意取4张纸币,则每种面值至少取到一张的概率为() A 8/91 B 25/91 C 48/91 D 53/91 E 60/91 F 63/91	算法 给定下列代码: <pre>for (int i=1; i<n; i*=3) { for (int j=i/3; j<i; j++) { Foo(); } }</pre> 已知n是一个整数:foo()时间复杂度为 $O(1)$,上述代码的时间复杂度是() A $O(\log n)$ B $O(n)$ C $O(n \cdot \log n)$ D $O(\log(n)^2)$			

屏幕剪辑的捕获时间: 2018/5/2 20:27

例1、商品推荐

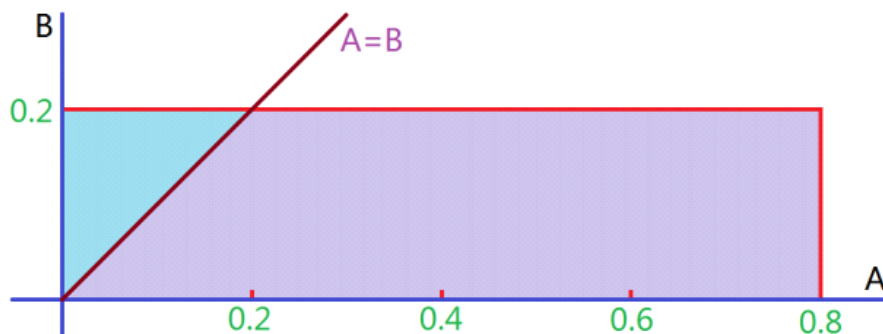
- ❑ 商品推荐场景中过于聚焦的商品推荐往往会损害用户的购物体验，在有些场景中，系统会通过一定程度的随机性给用户带来发现的惊喜感。
- ❑ 假设在某推荐场景中，经计算A和B两个商品与当前访问用户的匹配度分别为0.8分和0.2分，系统将随机为A生成一个均匀分布于0到0.8的最终得分，为B生成一个均匀分布于0到0.2的最终得分，试计算最终B的分数大于A的分数概率。

解法：算面积

屏幕剪辑的捕获时间: 2018/5/2 20:27

商品推荐

- ❑ A=B的直线上方区域，即为 $B > A$ 的情况。
- ❑ $S_{\text{蓝色}} = 0.02$ $S_{\text{矩形}} = 0.16$
- ❑ $p = 0.02 / 0.16 = 0.125$



屏幕剪辑的捕获时间: 2018/5/2 20:28

思考题：计算概率

- A、B两国元首相约在首都机场晚20点至24点交换一份重要文件。如果A国的飞机先到，A会等待1个小时；如果B国的飞机先到了，B会等待2个小时。假设两架飞机在20点至24点降落机场的概率是均匀分布，试计算能够在20点至24点完成交换的概率。
- 假设交换文件本身不需要时间。

屏幕剪辑的捕获时间: 2018/5/2 20:28

画图：

例2、进制推断

- 如果公式 $84 * 148 = B6A8$ 成立，则该公式采用的是_____进制表示的。
- A: 15
 - B: 11
 - C: 12
 - D: 14
 - E: 16

屏幕剪辑的捕获时间: 2018/5/2 20:28

常规：写出等式

84*148=B6A8

□ 1、常规做法：

□ 假定数值是x进制的，则写出等式：

$$(8x + 4) \cdot (x^2 + 4x + 8) = 11x^3 + 6x^2 + 10x + 8$$

$$\Rightarrow (3x^2 + 6x + 2) \cdot (x - 12) = 0$$

$$\Rightarrow x = 12$$

屏幕剪辑的捕获时间: 2018/5/2 20:28

优化的解法：

84*148=B6A8

□ 2、“启发式”做法：

□ 在十进制体系下，左侧个位乘积 $4*8=32$ ；

□ 右侧个位为8，差 $32-8=24$ ，从而进制必然是24的约数。只有C选项12是24的约数。

■ 以上分析，用“十进制体系”仅仅是计算习惯。

■ 这种做法可以辅助第1种解法做交互验证。

屏幕剪辑的捕获时间: 2018/5/2 20:28

例3、N个数的差

□ 将1,2,3,.....,99,100的任意排列成一个圈，则相邻两数的差的绝对值求和最大是多少？

- A: 100
- B: 198
- C: 200
- D: 500
- E: 2500
- F: 5000

屏幕剪辑的捕获时间: 2018/5/2 20:28

构造法

□ 定义：1-50为“小数”，51-100为“大数”，将“相邻元素的差的绝对值之和”简称“绝差之和”。

□ 贪心：为了将绝差之和取最大，应该避免将“大数”和“大数”放在一起。因此，可以考虑间隔排布。形成

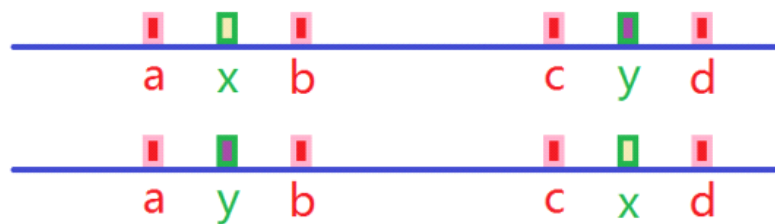
1,100,2,99,3,98,4,97.....,50,51的序列。而事实上，这个序列即绝差之和最大的序列。

屏幕剪辑的捕获时间: 2018/5/2 20:29

99+98+ 97+96+95+...+1.
= 100*50

小数和小数交换

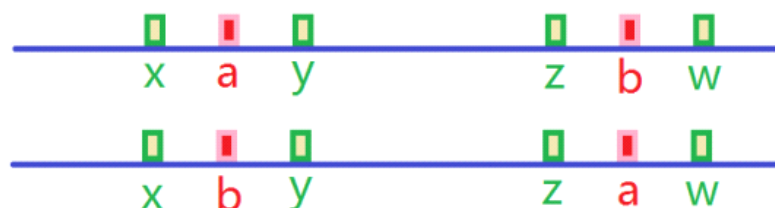
- 将小数 x 和 y 互换。假定交换之前， x 和 y 的相邻元素分别为 a 、 b 和 c 、 d 。
- 交换之前的绝差为： $(a-x)+(b-x)+(c-y)+(d-y)$
- 交换之后的绝差为： $(a-y)+(b-y)+(c-x)+(d-x)$
- 二者相等。



屏幕剪辑的捕获时间: 2018/5/2 20:29

大数和大数交换

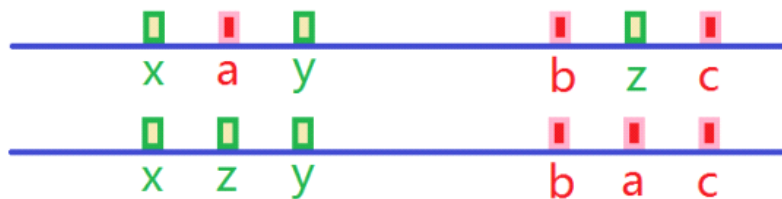
- 将大数 a 和 b 互换。假定交换之前， a 和 b 的相邻元素分别为 x 、 y 和 z 、 w 。
- 交换之前的绝差为： $(a-x)+(a-y)+(b-z)+(b-w)$
- 交换之后的绝差为： $(b-x)+(b-y)+(a-z)+(a-w)$
- 二者相等。



屏幕剪辑的捕获时间: 2018/5/2 20:29

大数和小数交换

- 将小数a和大数z互换。假定交换之前，a和z的相邻元素分别为x、y和b、c。
- 交换之前的绝差为： $(a-x)+(a-y)+(b-z)+(c-z)$
- 交换之后的绝差为： $|x-z|+|y-z|+|b-a|+|c-a|$
- 由于x、y、z更接近，a、b、c更接近，所以交换后绝差变小。



屏幕剪辑的捕获时间: 2018/5/2 20:29

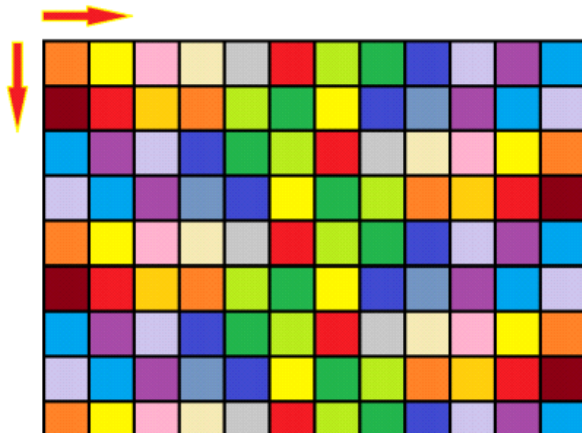
综上

- 无论如何交换，绝差之和都无法增大。即原序列1,100,2,99,3,98,4,97.....,50,51的绝差之和是最大的。它形成的绝差序列是99,98,97,96.....,2,1,50，这100个数的和为5000。
- 从“小小交换”和“大大交换”可以看到，绝差之和最大的序列不唯一。

屏幕剪辑的捕获时间: 2018/5/2 20:29

例4、走棋盘/格子取数

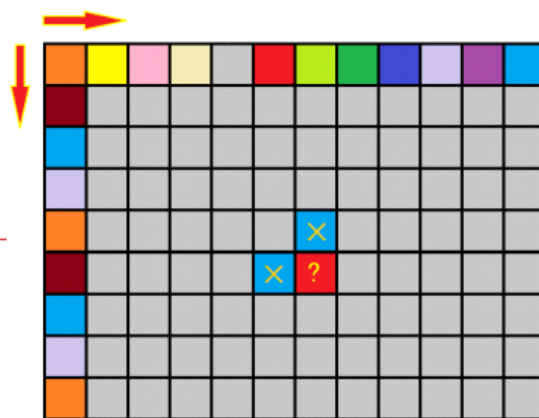
- 给定 $m \times n$ 的矩阵，每个位置是一个非负整数，从左上角开始，每次只能朝右和下走，走到右下角，求总和最小的路径。



屏幕剪辑的捕获时间: 2018/5/2 20:29

状态转移方程

- 走的方向决定了同一个格子不会经过两次。
- 若当前位于 (x,y) 处，它来自于哪些格子呢？
 - $dp[0,0]=a[0,0]$ / 第一行(列)累积
 - $dp[x,y] = \min(dp[x-1,y]+a[x,y], dp[x,y-1]+a[x,y])$
 - 即: $dp[x,y] = \min(dp[x-1,y], dp[x,y-1]) + a[x,y]$
- 思考：若将上述问题改成“求从左上到右下的最大路径”呢？

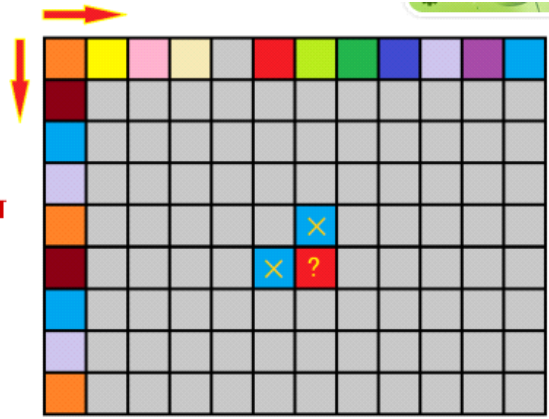


屏幕剪辑的捕获时间: 2018/5/2 20:29

状态转移方程

□ 状态转移方程：

$$\begin{cases} dp(i,0) = \sum_{k=0}^i chess[k][0] \\ dp(0,j) = \sum_{k=0}^j chess[0][k] \\ dp(i,j) = \min(dp(i-1,j), dp(i,j-1)) + chess[i][j] \end{cases}$$



□ 滚动数组：

$$\begin{cases} dp(j) = \sum_{k=0}^j chess[0][k] \\ dp(j) = \min(dp(j), dp(j-1)) + chess[i][j] \end{cases}$$

屏幕剪辑的捕获时间: 2018/5/2 20:30

```
int MinPath(vector<vector<int>> &chess, int M, int N)
{
    vector<int> pathLength(N);
    int i, j;

    //初始化
    pathLength[0] = chess[0][0];
    for (j = 1; j < N; j++)
        pathLength[j] = pathLength[j-1] + chess[0][j];

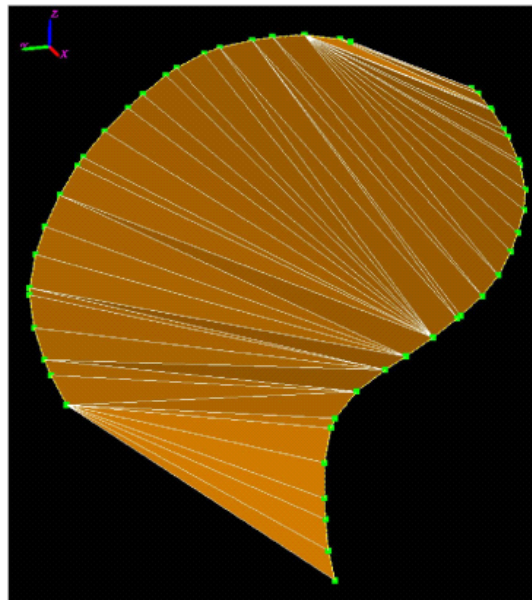
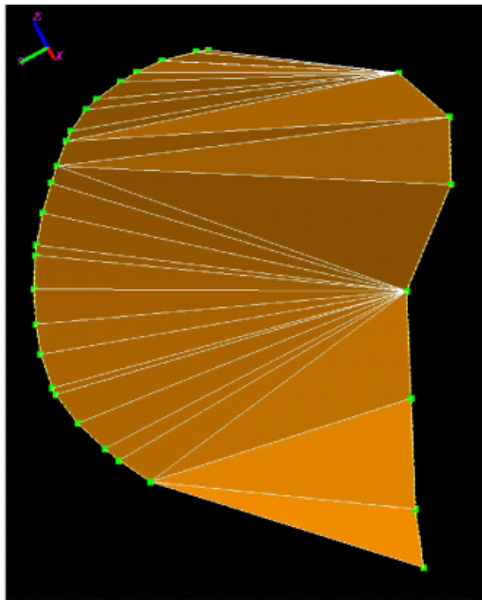
    //依次计算每行
    for (i = 1; i < M; i++)
    {
        pathLength[0] += chess[i][0];
        for (j = 1; j < N; j++)
        {
            if (pathLength[j-1] < pathLength[j])
                pathLength[j] = pathLength[j-1] + chess[i][j];
            else
                pathLength[j] += chess[i][j];
        }
    }
    return pathLength[N-1];
}

int _tmain(int argc, _TCHAR* argv[])
{
    const int M = 10;
    const int N = 8;
    vector<vector<int>> chess(M, vector<int>(N));

    //初始化棋盘: (随机给定)
    int i, j;
    for (i = 0; i < M; i++)
    {
        for (j = 0; j < N; j++)
            chess[i][j] = rand() % 100;
    }
    cout << MinPath(chess, M, N) << endl;
    return 0;
}
```

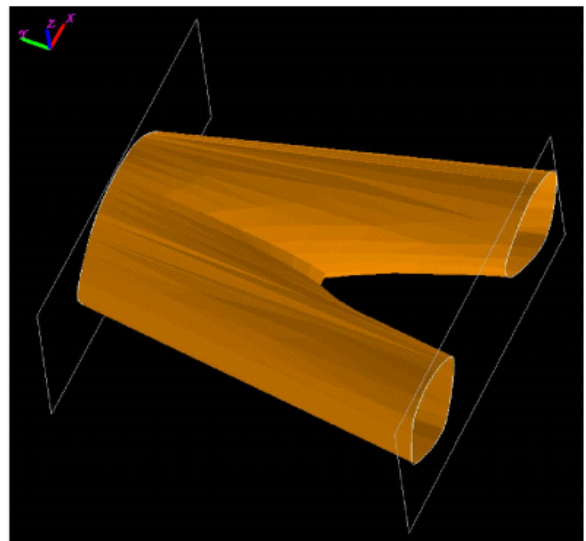
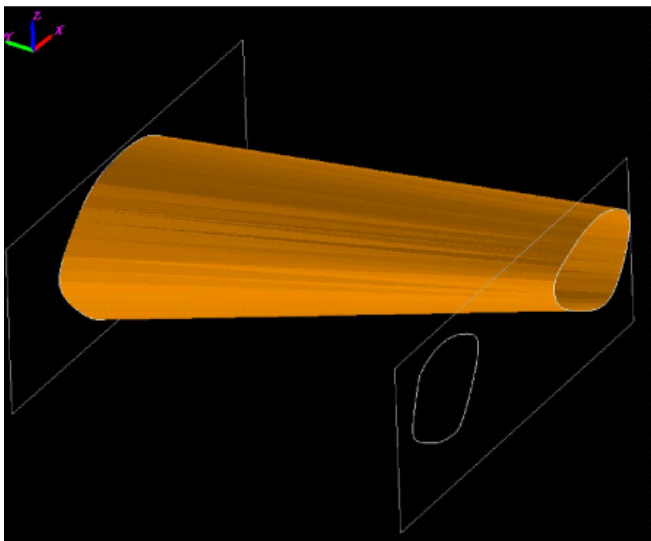
屏幕剪辑的捕获时间: 2018/5/2 20:30

实践：GIS中的应用



屏幕剪辑的捕获时间: 2018/5/2 20:30

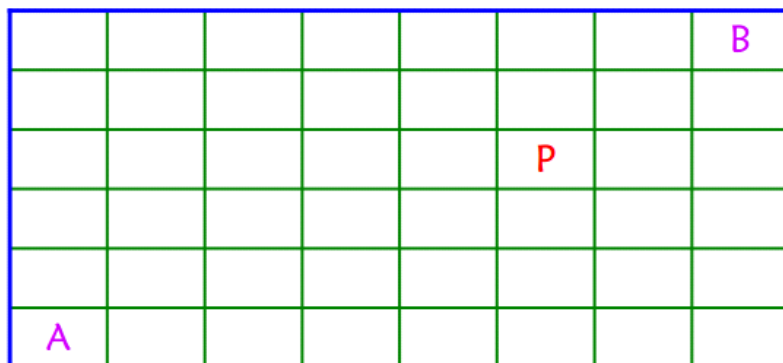
如果三维曲线是封闭线...



屏幕剪辑的捕获时间: 2018/5/2 20:30

例4.1、陷阱走棋盘

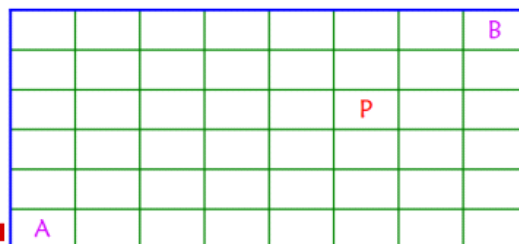
- 在 8×6 的矩阵中，每次只能向上或向右移动一格，并且不能经过P。试计算从A移动到B一共有多少种走法。



屏幕剪辑的捕获时间: 2018/5/2 20:30

计算a到p，计算p到b，总的减去两者相乘即可。

解题过程



- 从A到B共需要移动12步，其中7步向右，5步向上，可行走法数目为 $C_{12}^5 = 792$
- 从A到P共需要8步，其中5步向右，3步向上，可行走法数目为 $C_8^5 = 56$
- 从P到B共需要4步，其中2步向右，2步向上，可行走法数目为 $C_4^2 = 6$
- 则，从A到B经过P的路线有 $56 \times 6 = 336$ 种；
- 从A到B不经过P的路线有 $792 - 336 = 456$ 种。

屏幕剪辑的捕获时间: 2018/5/2 20:31

方格的可行路径数目

							B
					P		
A							

1	6	21	56	126	196	294	456
1	5	15	35	70	70	98	162
1	4	10	20	35	0	28	64
1	3	6	10	15	21	28	36
1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1

屏幕剪辑的捕获时间: 2018/5/2 20:31

例5、寻找程序员

□ A、B、C、D四人应聘一个程序员职位，此职务的要求条件是：Java熟练；懂数据库开发；会web开发；有C++经验。谁满足的条件最多，谁就被雇用。把这四个要求条件两两组合，每个组合都恰有一人满足。已知：

- A和B Java熟练
- B和C会web
- C和D懂数据库
- D有C++经验

□ 那么，被雇用的应该谁？

屏幕剪辑的捕获时间: 2018/5/2 20:31

整理信息

	Java	数据库	Web	C++
A	✓			
B	✓		✓	
C		✓	✓	
D		✓		✓

□ 根据条件得到表1:

□ 同时得到两两组合:

- 条件(甲): Java & Web: B
- 条件(乙): 数据库 & Web: C
- 条件(丙): 数据库 & C++: D
- 条件(丁): Web & C++: 待定
- 条件(戊): Java & C++: 待定
- 条件(己): Java & 数据库: 待定

屏幕剪辑的捕获时间: 2018/5/2 20:31

第一行 首先由甲知java, web一起是B的, 那么A既然会java, 那么定不会web

第二行c会数据库和web, 而b会web, 所以定不会数据库

第三行, d会数据库和c++, c会数据库了, 所以定不会c++. 然后b会java和web, 那么c会web, 则定不会java,

最后一行, 由条件乙, c会web和数据库, 而d有数据库, 那么不会web

此时, web这一项我们确定了。因此, 丁web和c++, 明显只能是B

同时java+c++也是B

那么D不会java, 否则重复, A也不会c++, 否则重复

屏幕剪辑的捕获时间: 2018/5/2 20:31

继续分析表2

	Java	数据库	Web	C++
A	✓		×(甲)	
B	✓	×(乙)	✓	✓
C	×(甲)	✓	✓	×(丙)
D		✓	×(甲)	✓

- 通过表2可知，
- Web 只有B和C两人掌握，
- 因此，“条件(丁): Web & C++”只能在这二人中产生，而C不会C++，所以，条件(丁)由B掌握。
- B已经掌握了Java，从而，“条件(戊): Java & C++”也是B掌握。
- D不能会Java，否则，D将与条件(戊)矛盾；
- A不能会C++，否则，A将与条件(戊)矛盾。
- 这时，B不会数据库，C、D不会Java，
- 所以，条件(己)只能有A掌握。

屏幕剪辑的捕获时间: 2018/5/2 20:31

最终结果

- 可以看到，B掌握了3项技术，而其他只掌握了两项技术。因此，被雇佣的应该是B。

	Java	数据库	Web	C++
A	✓	✓	×	×
B	✓	×	✓	✓
C	×	✓	✓	×
D	×	✓	×	✓

屏幕剪辑的捕获时间: 2018/5/2 20:31

附：谁是情种？

- 李逵、宝玉、西门、纳兰四才子穿越到女儿国，结识了西施、貂蝉、昭君、玉环。国王武氏将选择一个情种做贴身侍卫。武氏发现，任选两美人，恰只有一位才子喜欢他们俩。狄仁杰已探得：
 - 李逵、宝玉喜欢西施
 - 西门、纳兰喜欢貂蝉
 - 宝玉、西门喜欢昭君
 - 纳兰喜欢玉环
- 未知情报需要你来帮助狄仁杰推断，请问，谁能够成为武氏的贴身侍卫？
 - 不知情种是何物者，先面壁七十二时辰！

屏幕剪辑的捕获时间: 2018/5/2 20:32

例6、古典概型

- 现有一叠纸币，其中5元面值的纸币6张，10元面值的纸币5张，20元面值的纸币4张，从袋子中任意取4张纸币，则每种面值至少取到一张的概率是多少？

屏幕剪辑的捕获时间: 2018/5/2 20:32

问题分析

- 现有一叠纸币，其中5元面值的纸币6张，10元面值的纸币5张，20元面值的纸币4张，从袋子中任意取4张纸币，则每种面值至少取到一张的概率是多少？
- 问题分析：基本事件总数为从15张纸币中取4张的所有取法，有效事件为满足题意的取法。

屏幕剪辑的捕获时间: 2018/5/2 20:32

事件总数为 $c(15,4)$

有效事件有 从5,10,20三种中任取1种取两张，其余取一张

$C_6^2 \times 2 \times 5 + C_5^2 \times 2 \times 4 + C_4^2 \times 2 \times 5$

问题分析

- 题干：6张/5张/4张 - 任取4张纸币
- 基本事件总数： C_{15}^4
- 有效事件分三种情况：从5元、10元和20元面值中选其中一种面值取2张，其他两种面值各取1张。总事件数目为：
$$C_6^2 \times 5 \times 4 + C_5^2 \times 6 \times 4 + C_4^2 \times 6 \times 5$$
- 上面两式相除即为概率，化简后得到48/91。

屏幕剪辑的捕获时间: 2018/5/2 20:32

例7、利用古典概型求概率与期望

- 从1,2,3,.....,99,2015里任意选择一部分数(可能为0个数), 这部分数按位异或的期望值是多少?
- A: 512
 - B: 1007
 - C: 1008
 - D: 2015/2
 - E: 1024
 - F: 2047/2

屏幕剪辑的捕获时间: 2018/5/2 20:32

异或结果与二进制位1的个数为奇数还是偶数有关, 和0个数无关

2015,11位,

问题分析

- 针对任何一个二进制位: 取奇数个1异或后会得到1, 取偶数个1异或后会得到0; 与取0的个数无关。
- 给定的最大数2015=(11111011111)₂, 共11位。针对每一位分别计算: 考虑第i位X_i:
- 假定给定的这些数中第i位一共有N个1, M个0, 某次采样取到的1的个数为k。则有:

$$P\{X_i = 1\} = \frac{2^m \cdot \sum_{k \in \text{odd}} C_n^k}{2^{m+n}} = \frac{\sum_{k \in \text{odd}} C_n^k}{2^n} = \frac{1}{2}$$

- 第一个等式: 从N个数中选K个数, K取奇数的可能性
- 第二个等式: 组合数奇数项、偶数项的各自加和相等

屏幕剪辑的捕获时间: 2018/5/2 20:33

利用期望公式 $E(X+Y)=E(X)+E(Y)$

□ 从上式得到：11位二进制数中，每个位取1的期望都是1/2，从而：

$$\begin{aligned} E(X) &= E\left(\sum_{i=1}^{11} (X_i \cdot P\{X_i\})\right) \\ &= E\left(\sum_{i=1}^{11} (2^i \cdot P\{X_i = 1\} + 0 \cdot P\{X_i = 0\})\right) \\ &= E\left(\sum_{i=1}^{11} (2^i \cdot P\{X_i = 1\})\right) \\ &= \sum_{i=1}^{11} E(2^i \cdot P\{X_i = 1\}) = \sum_{i=1}^{11} 2^i \cdot E(P\{X_i = 1\}) \\ &= \sum_{i=1}^{11} 2^i \cdot \frac{1}{2} = \frac{1}{2} \sum_{i=1}^{11} 2^i = \frac{(11111111111)_2}{2} \\ &= 1023.5 \end{aligned}$$

屏幕剪辑的捕获时间: 2018/5/2 20:33

例8、时间复杂度

□ 假定函数MyFunc()的时间复杂度为 $O(1)$ ，则下列代码的时间复杂度关于整数n是多少？

■ $\Theta(N \log N)$ / $\Theta(N)$

□ 注： Θ 表示复杂度是紧的，

□ 如堆排序中，建堆的时间复杂度为 $\Theta(N)$ ，而非 $\Theta(N \log N)$ ；

□ 当然，可以说建堆的时间复杂度为 $O(N \log N)$ ，因为O记号不要求上确界。

```
void CalcTime()
{
    int i, j;
    for (i = 1; i < n; i *= 3)
    {
        for (j = i/3; j < i; j++)
        {
            MyFunc();
        }
    }
}
```

屏幕剪辑的捕获时间: 2018/5/2 20:33

等比数列求和，以及 3^k 是小于N的，看通项那里

时间复杂度分析

- 内层循环中，对于给定的 i ， j 从 $\frac{i}{3}$ 累加到 i ，循环次数为 $\frac{2}{3} \cdot i$
- 外层循环中， i 从1到 n 遍历，每次变成当前值的3倍，即1,3,9,27.....，通项为 $3^k (k = 0, 1, 2 \dots, 3^k < N)$
- 将内层循环次数按照递增3倍做累加后，得循环总次数：
$$\begin{aligned} Time &= \frac{2}{3} \cdot 1 + \frac{2}{3} \cdot 3 + \frac{2}{3} \cdot 9 + \frac{2}{3} \cdot 27 + \frac{2}{3} \cdot 81 + \dots + \frac{2}{3} \cdot 3^k \\ &= \frac{2}{3} \cdot (1 + 3 + 9 + 27 + \dots + 3^k) \\ &= \frac{2}{3} \cdot \frac{1 - 3^{k+1}}{1 - 3} = 3^k - \frac{1}{3} < N \end{aligned}$$

屏幕剪辑的捕获时间: 2018/5/2 20:33

图示分析

- 从下面的图示能够清楚的反映这一问题：



- 上图中，当外层循环的 i 位于紫色位置时，内层循环执行的是紫色的①；下次循环，当外层循环的 i 位于红色位置 $3*i$ 时，内层循环执行的是红色的②，依次类推。所以，循环次数的上限为 N 。从而，时间复杂度为 $O(N)$ 。

屏幕剪辑的捕获时间: 2018/5/2 20:34

思考题1

- 猜数字：游戏的过程是你输入一个4位数(数字选取0~9，不会重复)，电脑会告诉你两个信息：A个数字和位置均正确，B个数字正确但位置错误。
- 例如正确答案是7890，你输入0789，电脑会告诉你0A4B，表示有0个数字和位置完全正确，但有4个数字正确但位置错误。
- 现在小猿猜了三次的结果分别是：1234=>0A4B, 2341=>1A3B, 3412=>2A2B，那么小猿最少还要猜几次才能保证一定得到4A0B的结果？

经过分析，只有四种情况

2 4 1 3

3 1 4 2

3 4 2 1

4 3 1 2

现在考虑是否可以询问一次就得到答案。

A+B的总数量肯定无法提供额外的信息。

所以我们只能通过0A, 1A, 2A, 3A, 4A来获取信息。

如果达到3A，则只能至少猜一个1, 2, 3, 4以外的数字。

这样的话，这个位置就废掉了，没啥意义。

所以我们只能通过0A, 1A, 2A, 4A来区分四种情况

如果用4A, 如猜2 4 1 3

2 4 1 3 => 2A

3 1 4 2 => 0A

3 4 2 1 => 1A

4 3 1 2 => 1A

无法区分两个1A的情况，由对称性，猜其他三个类似。

所以我们至少再猜两次才能知道答案。

所以我们至少再猜三次才能保证得到4A

屏幕剪辑的捕获时间: 2018/5/2 20:34

思考题2

- 2015年年初，某公司部门举行年会，期间有个游戏，规则是通过猜拳的方法决出每一局的胜负(一人赢其他所有人才算赢得本局，出现平局就继续猜拳)，如果谁先赢满7局，则获胜，并赢得全部奖金，现在有3名员工甲、乙、丙参与了这个游戏，游戏进行了一会，分数为甲6，乙5，丙4，但时间已晚，大家要吃饭了，最公平的奖金分配方案是_____
- A. 甲得到19/27，乙得到7/27，丙得到1/27
 - B. 甲得到18/27，乙得到8/27，丙得到1/27
 - C. 甲得到17/27，乙得到7/27，丙得到1/27
 - D. 甲得到19/27，乙得到6/27，丙得到2/27
 - E. 甲得到18/27，乙得到7/27，丙得到2/27
 - F. 甲得到17/27，乙得到6/27，丙得到2/27

屏幕剪辑的捕获时间: 2018/5/2 20:34

回答:

甲赢: 需要1轮才能结束游戏: 甲;

需要2轮才能结束游戏 (丙1次, 否则乙1次): 乙甲、丙甲;

需要3轮才能结束游戏 (丙1次且乙1次): 丙乙甲、乙丙甲;

需要4轮才能结束游戏 (丙2次且乙1次):

乙赢: 6/27

需要2轮才能结束游戏: $1/3 * 1/3$

需要3轮才能结束游戏 (丙在前两轮任意赢得1次, 乙1次): $2 * 1/3 * 1/3 * 1/3$

需要4轮才能结束游戏 (丙在前三轮任意赢2次, 乙1次): $3 * 1/3 * 1/3 * 1/3 * 1/3$

丙赢: 2/27

需要3轮才能结束游戏: $1/3 * 1/3 * 1/3$

需要4轮才能结束游戏 (丙在前三轮任意赢2次, 乙1次): $3 * 1/3 * 1/3 * 1/3 * 1/3$

简单枚举答案结果的话, 接下来比赛赢的序列中, 乙赢的可能有{乙乙, 丙乙乙, 乙丙乙, 乙丙丙乙, 丙乙丙乙, 丙丙乙乙}=1/9+1/27*2+1/81*3=6/27, 丙赢的可能有{丙丙丙, 乙丙丙丙, 丙乙丙丙, 丙丙乙丙}=1/27+1/81*3=2/27, 甲的概率也就是19/27了; 正常标准的计算步骤差不多也就是枚举某人赢的可能场次, 遇到比较大的数字时用组合数的公式去算

思考题3

- 甲乙两个人比试射箭，两人射术水平一样，如果甲射了101箭，而乙射了100箭，甲射中次数比乙射中次数多的概率是多少？

屏幕剪辑的捕获时间: 2018/5/2 20:34

链接:

<https://www.nowcoder.com/questionTerminal/cb32d803d4314d9293f0310227a88d15?orderByHotValue=0&done=0&pos=15&mutiTagIds=601&onlyReference=false>

然后前一百次可以分为三种情况：甲多、乙多、一样多；因为水平一样，所以甲多、乙多的概率相等；

因为射中概率为0.5，所以前一百次一样多并且最后一次甲射中的概率就是前一百次一样多的概率/2（即*0.5）；

最后甲多的概率 = 前一百次甲多的概率 + 前一百次一样多并且最后一次甲射中的概率
= 前一百次（甲多+乙多）的概率/2 + 前一百次一样多的概率/2
= 前一百次（甲多+乙多+一样多）的概率/2
= 1/2

屏幕剪辑的捕获时间: 2018/5/2 21:22

因为2人水平一样，则前100次2人平手，第101次是关键，甲只有两种情况，射中和不射中（概率都是1/2），射中的话甲比乙多，不射中的话相等

来自 <<https://www.nowcoder.com/questionTerminal/cb32d803d4314d9293f0310227a88d15?orderByHotValue=0&done=0&pos=15&mutiTagIds=601&onlyReference=false>>

一点总结

☐ 重视思想，重视动手能力

- 计算机是实践学科，掌握至少一门编程语言。
- 提高内涵是赢取所有对战平台的终极法宝。

☐ 重视教科书上的基本内容

- 实现无bug的快速排序多少分钟？
- 广度优先搜索用什么辅助数据结构？
- 字符串的全排列怎么设计算法及代码实现？
 - ☐ 字符有重复怎么办？
 - ☐ 递归？非递归？

屏幕剪辑的捕获时间: 2018/5/2 20:34