

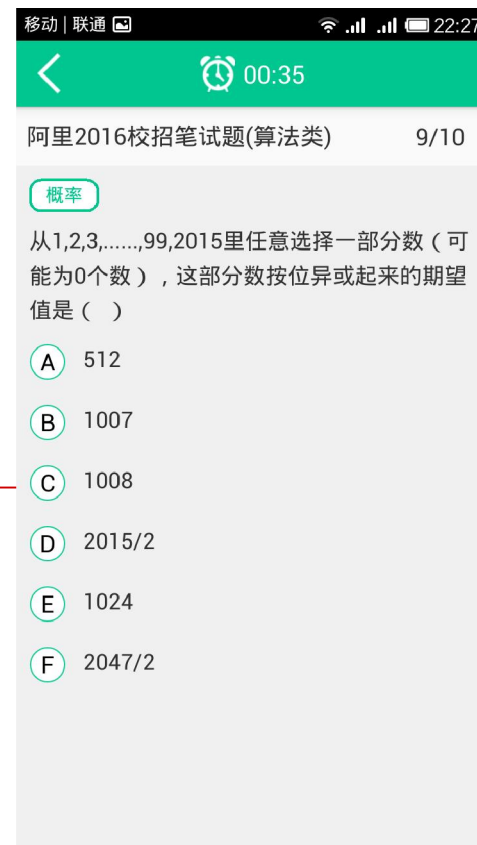
助力2016春招

题 七月题库
笔试面试刷题神器

©北京七月在线科技有限公司

7 | 七月算法 | 邹博
julyedu.com

2016年2月28日



主要内容

- 古典概型
 - 找零钱/麻将
- 众数
 - 绝对众数
- 链表(部分)逆序
 - 附：链表去重及其变种
- 金钗赛诗II
 - 带约束的排序
- 附：猜数字
 - 面试常考类型——逻辑思维能力
- 附：跳跃问题
 - 贪心/动态规划/广度优先搜索



即将进行的...

□ 3月面试求职班

■ 3月8日起每周二、四、六晚20-22点，一周三课

□ 4月机器学习班

■ 4月9日起每周六、日晚19-21点，一周双课

4、数组

Top10问题寻优

最大连续子数组

X-Sum (子集和数问题): 2-sum、n-sum等

N指针方法: 奇偶排序、荷兰国旗等

深化循环不变式

5、树

二叉搜索树删除

二叉树的遍历 (递归、非递归)

详解平衡二叉树 (AVL树)

2-3-4树、红黑树

B树/R树

6、图

Bellman-Ford算法/Dijkstra算法/Floyd算法

最小生成树 (Prim算法、Kruskal算法)

实践应用: Word-Ladder/马踏棋盘/数独问题/N-皇后问题

7、查找排序

无bug二分查找的实现与技巧

行列递增杨氏矩阵的查找

归并排序/快速排序/堆排序及其应用

基数排序/计数排序/桶排序及其应用

8、贪心和动态规划

从机器学习的角度统一贪心法和动态规划

贪心的应用: 任务安排问题、甘特图

走棋盘-及其实践中的重要应用

典型DP应用: 格子取数问题/字符串的交替连接/矩阵连乘/背包问题

数零钱问题/回文划分问题/单词划分问题

9、概率组合数论统计

现实中的概率问题: 以麻将为例

用动态规划推导组合公式

圆内/任意多边形均匀取点问题

歌曲、商品等推荐问题

金钗赠诗问题/树型状态

10、BAT 海量数据处理、系统设计题精讲

Darts高效索引

Hash和树的综合: POI问题

详解跳跃表/Bloom Filter

遗传算法/蚁群算法

第9课 推荐系统

协同过滤、隐语义模型pLSA/SVD、随机游走Random Walk

实践示例: 协同过滤代码实现和参数调试分析

第10课 SVM

线性可分支持向量机、线性支持向量机、非线性支持向量机、SMO

实践案例: 使用SVM进行数据分类 [含代码实现和参数调试分析]

第11课 贝叶斯网络

朴素贝叶斯、有向分离、马尔科夫模型/HMM/pLSA

第12课 EM算法

GMM、pLSA、HMM

实践案例: 分解男女身高、图像分割

第13课 主题模型

pLSA、共轭先验分布、LDA

实践案例: 使用LDA进行文档聚类 [含代码实现和参数调试分析]

第14课 采样与变分

MCMC/KL(p||q)与KL(q||p)

第15课 隐马尔可夫模型HMM

概率计算问题、参数学习问题、状态预测问题

实践案例: 使用HMM进行中文分词 [含代码实现和参数调试分析]

第16课 条件随机场CRF

概率无向图模型、MRF、线性链CRF

第17课 人工神经网络

全连接神经网络、BP算法、链式法则

实践案例: 用BP网络做样本数据分类

第18课 深度学习之CNN

卷积操作、Pooling操作

理解卷积网络、卷积网络训练以及GPU加速

卷积网络在图像分类中的应用

第19课 深度学习之RNN

RNN网络、LSTM

回归网络在自然语言处理中的应用: 字符模型和文本生成, char-rnn 案例

第20课 深度学习实践

Caffe 入门教程

Tensor Flow/MxNet 简介

实践案例: 使用Torch进行图像分类及卷积网络可视化的深度学习实践



2016春招公开课

例1、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？



问题分析

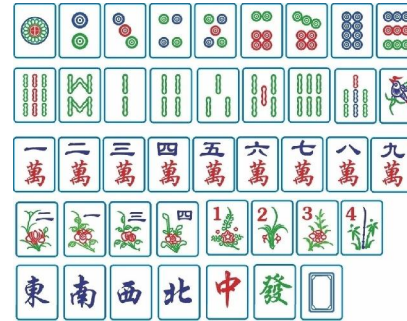
- 现有一叠纸币，其中5元面值的纸币6张，10元面值的纸币5张，20元面值的纸币4张，从袋子中任意取4张纸币，则每种面值至少取到一张的概率是多少？
- 问题分析：基本事件总数为从15张纸币中取4张的所有取法，有效事件为满足题意的取法。



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。





例1.1、身边的概率

- “国粹”麻将是集技巧与运气的在我国开展广泛的娱乐项目。去除花牌后的标准麻将，由1到9的“万、条、饼”各4张，以及“东南西北中发白”各4张，共计136张组成。我们把两张内容一样的牌叫一幅“将”。
- 请问，庄家起手摸14张牌，则他起手没有“将”的概率是多少？
 - 此外，可以算下摸13张牌没有“将”的概率，摸13张牌没有“风”的概率。



问题分析

□ 基本事件数目：

■ 一共136张牌，随意选4张，取法为： C_{136}^{14}

□ 有效事件数目：

■ 一共34组牌，选择某个组，然后在该组的4张中任选1张，取法为： $C_{34}^{14} \cdot 4^{14}$

□ “无将”的概率为：

$$p = \frac{C_{34}^{14} \cdot 4^{14}}{C_{136}^{14}} = \frac{[21 \sim 34] \cdot 4^{14}}{[123 \sim 136]} \approx 8.79\%$$



例2、绝对众数

- 定义：给定 N 个数，称出现次数最多的数为众数；若某众数出现的次数大于 $N/2$ ，称该众数为绝对众数。
 - 如： $A=\{1,2,1,3,2\}$ 中，1和2都是众数，但都不是绝对众数。
 - 如： $A=\{1,2,1,3,1\}$ 中，1是绝对众数。
- 已知给定的 N 个整数存在绝对众数，以最低的是空复杂度计算该绝对众数。



算法分析

- 删除数组A中两个不同的数，绝对众数不变。
 - 若两个数中有1个是绝对众数，则剩余的N-2个数中，绝对众数仍然大于 $(N-2)/2$ ；
 - 若两个数中没有绝对众数，显然不影响绝对众数。
- 算法描述：
- 记m为候选绝对众数，出现次数为c，初始化为0。
- 遍历数组A：
 - 若 $c==0$ ，则 $m=A[i]$
 - 若 $c \neq 0$ 且 $m \neq A[i]$ ，则同时删掉m和A[i]
 - 若 $c \neq 0$ 且 $m==A[i]$ ，则 $c++$



Code

```
int Mode(int* a, int size)
{
    int count = 0;
    int m = a[0];
    for(int i = 0; i < size; i++)
    {
        if(count == 0)
        {
            m = a[i];
            count = 1;
        }
        else if(m != a[i])
        {
            count--;
        }
        else //if(m == a[i])
        {
            count++;
        }
    }
    return m;
}

int _tmain(int argc, _TCHAR* argv[])
{
    int a[] = {8, 8, 1, 1, 1, 8, 1, 1, 6, 1, 8};
    int m = Mode(a, sizeof(a)/sizeof(int));
    cout << m << endl;
    return 0;
}
```



例3、链表的部分翻转

- 给定一个链表，翻转该链表从m到n的位置。
要求直接翻转而非申请新空间。
- 如：给定 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ ， $m=2$ ， $n=4$ ，返回 $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 5$ 。
- 假定给出的参数满足： $1 \leq m \leq n \leq \text{链表长度}$ 。



分析

- 空转 $m-1$ 次，找到第 $m-1$ 个结点，即开始翻转的第一个结点的前驱，记做head；
- 以head为起始结点遍历 $n-m$ 次，将第 i 次时，将找到的结点插入到head的next中即可。
 - 即头插法



Code

64 → 62 → 58 → 78 → 24 → 69 → 0 → 34 → 67 → 41

64 → 62 → 58 → 34 → 0 → 69 → 24 → 78 → 67 → 41

```

typedef struct tagSNode
{
    int value;
    tagSNode* pNext;

    tagSNode(int v) : value(v), pNext(NULL) {}
} SNode;

int _tmain(int argc, _TCHAR* argv[])
{
    SNode* pHead = new SNode(0);
    int i;
    for(i = 0; i < 10; i++)
    {
        SNode* p = new SNode(rand() % 100);
        p->pNext = pHead->pNext;
        pHead->pNext = p;
    }
    Print(pHead);
    Reverse(pHead, 4, 8);
    Print(pHead);
    Destroy(pHead);
    return 0;
}

void Destroy(SNode* p)
{
    SNode* next;
    while(p)
    {
        next = p->pNext;
        delete p;
        p = next;
    }
}

void Reverse(SNode* pHead, int from, int to)
{
    SNode* pCur = pHead->pNext;
    int i;
    for(i = 0; i < from-1; i++)
    {
        pHead = pCur;
        pCur = pCur->pNext;
    }

    SNode* pPre = pCur;
    pCur = pCur->pNext;
    to--;
    SNode* pNext;
    for(; i < to; i++)
    {
        pNext = pCur->pNext;
        pCur->pNext = pHead->pNext;
        pHead->pNext = pCur;
        pPre->pNext = pNext;
        pCur = pNext;
    }
}

```



附：排序链表去重

□ 给定排序的链表，删除重复元素，只保留重复元素第一次出现的结点。

□ 如：

■ 给定： $2 \rightarrow 3 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 8 \rightarrow 8 \rightarrow 9 \rightarrow 9 \rightarrow 10$

■ 返回： $2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10$



Code

```
typedef struct tagSNode
{
    int value;
    tagSNode* pNext;

    tagSNode(int v) : value(v), pNext(NULL) {}
} SNode;

int _tmain(int argc, _TCHAR* argv[])
{
    SNode* pHead = new SNode(0);
    int data[] = {2, 3, 3, 5, 7, 8, 8, 8, 9, 9, 30};
    int size = sizeof(data) / sizeof(int);
    for(int i = size-1; i >= 0; i--)
    {
        SNode* p = new SNode(data[i]);
        p->pNext = pHead->pNext;
        pHead->pNext = p;
    }
    Print(pHead);
    DeleteDuplicateNode(pHead);
    Print(pHead);
    Destroy(pHead);
    return 0;
}

void DeleteDuplicateNode(SNode* pHead)
{
    SNode* pPre = pHead->pNext;
    SNode* pCur;
    while(pPre)
    {
        pCur = pPre->pNext;
        if(pCur && (pCur->value == pPre->value))
        {
            pPre->pNext = pCur->pNext;
            delete pCur;
        }
        else
        {
            pPre = pCur;
        }
    }
}
```



Code2

□ 分析该代码的正确性

```
void DeleteDuplicateNode2(SNode* pHead)
{
    SNode* pPre = pHead;
    SNode* pCur = pPre->pNext;
    SNode* pNext;
    while(pCur)
    {
        pNext = pCur->pNext;
        while(pNext && (pCur->value == pNext->value))
        {
            pPre->pNext = pNext;
            delete pCur;
            pCur = pNext;
            pNext = pCur->pNext;
        }
        pPre = pCur;
        pCur = pNext;
    }
}
```



附：排序链表去重II

□ 若题目变成：若发现重复元素，则重复元素全部删除，代码应该怎么实现呢？

□ 如：

■ 给定：2→3→3→5→7→8→8→8→9→9→10

■ 返回：2→5→7→10



Code

```
void DeleteDuplicateNode3(SNode* pHead)
{
    SNode* pPre = pHead;
    SNode* pCur = pPre->pNext;
    SNode* pNext;
    bool bDup;
    while(pCur)
    {
        pNext = pCur->pNext;
        bDup = false;
        while(pNext && (pCur->value == pNext->value))
        {
            pPre->pNext = pNext;
            delete pCur;
            pCur = pNext;
            pNext = pCur->pNext;
            bDup = true;
        }
        if(bDup) //此刻的pCur与原数据重复，删之
        {
            pPre->pNext = pNext;
            delete pCur;
        }
        else //pCur未发现重复，则pPre后移
        {
            pPre = pCur;
        }
        pCur = pNext;
    }
}
```



例4、金钗赛诗II

- 赛诗会上十二金钗各赋诗一首，12人各自随机挑选一首后，李纨曰：“大家通过两两交换的方式，换回自己的律诗；但要求只能跟我交换”。现已知：

黛玉、宝钗、元春、探春、湘云、妙玉、
迎春、惜春、熙凤、巧姐、李纨、可卿

各自拿到的律诗作者为：

熙凤、黛玉、迎春、惜春、湘云、可卿、
探春、元春、宝钗、巧姐、妙玉、李纨

- 试计算至少需要多少次交换，才能使得所有人交换得到自己的律诗？



问题分析

- 十二金钗分别标号为 $0, 1, 2, \dots, 11$ ，其中，李纨为0号。题目实际上给定了12个整数的一个排列，要求只和0交换，最终形成升序数组，求最少的交换次数。
- 统计每个“环”的长度 L ：
 - 若包含0，则该环的最少交换次数为 $L-1$ ；
 - 若不包含0，则该环的最少交换次数为 $L+1$ ；
- 将所有环的交换次数累积即可。
 - 时间复杂度 $O(N)$ ，空间复杂度 $O(N)$



Code

```
int _tmain(int argc, _TCHAR* argv[])
{
    int N = 12;
    int* a = new int[N];
    for(int i = 0; i < N; i++)
        a[i] = i;
    random_shuffle(a, a+N);
    cout << Exchange0(a, N) << endl;
    delete[] a;
    return 0;
}
```

```
int Exchange0(const int* a, int size)
{
    bool* visit = new bool[size];
    memset(visit, 0, sizeof(bool)*size);
    int j;
    int c = (a[0] == 0) ? 2 : 0;
    for(int i = 0; i < size; i++)
    {
        if(visit[i] || (a[i] == i))
            continue;
        j = a[i];
        while(j != i)
        {
            c++;
            visit[j] = true;
            j = a[j];
        }
        c += 2;
    }
    c -= 2;
    delete[] visit;
    return c;
}
```

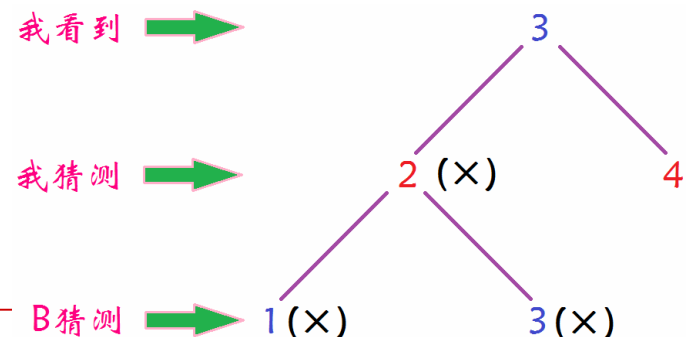


附：猜数字游戏

- 两个聪明人A和B玩猜数字的游戏。他们在脑门上各贴一个正整数数字，两个数字只相差1，A和B只能看到对方的数组而看不到自己的。
- 以下是两人的对话：
 - A：我不知道
 - B：我也不知道
 - A：我知道了
 - B：我也知道了
- 上述4句对话结束后，聪明的你帮助A、B推算下，他们的数字各是多少呢？



附：逻辑总结



- 复杂的逻辑题可以用**二叉树(N叉树)**做辅助推理，原理是：只要某个结点的两个孩子(所有孩子)都不可能，则这个结点不可能。
 - 如A4B3情况下，A的**推理树**如右上图所示。
- 注意：两人的说话顺序是有决定作用的，是**不对称消息**：
 - A说话，则A是在看到B的内容后**做判断**，B可根据A的内容在自己的推理树上**做剪枝**。



附：Jump

□ 跳跃问题

- 给定非负整数数组，初始时在数组起始位置放置一机器人，数组的每个元素表示在当前位置机器人最大能够跳跃的数目。它的目的是用最少的步数到达数组末端。例如：给定数组 $A=[2,3,1,1,2]$ ，最少跳步数目是2，对应的跳法是： $2 \rightarrow 3 \rightarrow 2$ 。
- 如： $2,3,1,1,2,4,1,1,6,1,7$ ，最少需要几步？



Code

```
int Jump(int A[], int n)
{
    if (n == 1)
        return 0;

    int step = 0; //最小步数
    int i = 0;
    int j = 0; // [i, j] 是当前能覆盖的区间
    int k, j2;
    while (j < n) //覆盖区间尚未包含最后元素
    {
        step++;
        j2 = j;
        for (k = i; k <= j; k++)
        {
            j2 = max(j2, k + A[k]);
            if (j2 >= n-1) //已经跳跃到最后一步
                return step;
        }
        i = j+1;
        j = j2;
        if (j < i) //覆盖区间为负，说明无法跳到末尾
            return -1;
    }
    return step;
}

int _tmain(int argc, _TCHAR* argv[])
{
    int A[] = {2, 3, 1, 1, 2, 4, 1, 1, 6, 1, 7};
    Jump(A, sizeof(A) / sizeof(int));
    return 0;
}
```



附： Jump问题“知识挖掘”

- 上述代码的时间复杂度是多少？
 - $O(N)$ or $O(N^2)$
- 该算法能够天然处理无法跳跃到末尾的情况。
 - 若无法跳到末尾，则返回-1
- 该算法在每次跳跃中，都是尽量跳的更远，并记录j2——属于贪心法；也可以认为是从区间[i,j](若干结点)扩展下一层区间[j+1,j2](若干子结点)——属于广度优先搜索。
 - 可见，贪心法是需要详细分析才能放心使用。
 - 回忆图论中的概要说明：
 - 广度优先搜索往往和“最少”、“最短”相关联。
- 思考：是否可以使用动态规划解决？
 - 记dp[i]为：到达A[i]时，还剩余多少步没有用。
 - 则：dp[i+1]=max(dp[i],A[i])-1



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的快速排序需要几分钟？

☐ 掌握内容要全面、透彻。

- 优先队列是什么，教科书上在哪部分介绍的？
- 计算字符串的全排列如何设计算法及代码实现？
 - ☐ 字符有重复怎么办？
 - ☐ 递归？非递归？



即将进行的...

□ 3月面试求职班

■ 3月8日起每周二、四、六晚20-22点，一周三课

□ 4月机器学习班

■ 4月9日起每周六、日晚19-21点，一周双课

4、数组

Top10问题寻优

最大连续子数组

X-Sum (子集和数问题): 2-sum、n-sum等

N指针方法: 奇偶排序、荷兰国旗等

深化循环不变式

5、树

二叉搜索树删除

二叉树的遍历 (递归、非递归)

详解平衡二叉树 (AVL树)

2-3-4树、红黑树

B树/R树

6、图

Bellman-Ford算法/Dijkstra算法/Floyd算法

最小生成树 (Prim算法、Kruskal算法)

实践应用: Word-Ladder/马踏棋盘/数独问题/N-皇后问题

7、查找排序

无bug二分查找的实现与技巧

行列递增杨氏矩阵的查找

归并排序/快速排序/堆排序及其应用

基数排序/计数排序/桶排序及其应用

8、贪心和动态规划

从机器学习的角度统一贪心法和动态规划

贪心的应用: 任务安排问题、甘特图

走棋盘-及其实践中的重要应用

典型DP应用: 格子取数问题/字符串的交替连接/矩阵连乘/背包问题

数零钱问题/回文划分问题/单词划分问题

9、概率组合数论统计

现实中的概率问题: 以麻将为例

用动态规划推导组合公式

圆内/任意多边形均匀取点问题

歌曲、商品等推荐问题

金钗赠诗问题/树型状态

10、BAT 海量数据处理、系统设计题精讲

Darts高效索引

Hash和树的综合: POI问题

详解跳跃表/Bloom Filter

遗传算法/蚁群算法

第9课 推荐系统

协同过滤、隐语义模型pLSA/SVD、随机游走Random Walk

实践示例: 协同过滤代码实现和参数调试分析

第10课 SVM

线性可分支持向量机、线性支持向量机、非线性支持向量机、SMO

实践案例: 使用SVM进行数据分类 [含代码实现和参数调试分析]

第11课 贝叶斯网络

朴素贝叶斯、有向分离、马尔科夫模型/HMM/pLSA

第12课 EM算法

GMM、pLSA、HMM

实践案例: 分解男女身高、图像分割

第13课 主题模型

pLSA、共轭先验分布、LDA

实践案例: 使用LDA进行文档聚类 [含代码实现和参数调试分析]

第14课 采样与变分

MCMC/KL(p||q)与KL(q||p)

第15课 隐马尔可夫模型HMM

概率计算问题、参数学习问题、状态预测问题

实践案例: 使用HMM进行中文分词 [含代码实现和参数调试分析]

第16课 条件随机场CRF

概率无向图模型、MRF、线性链CRF

第17课 人工神经网络

全连接神经网络、BP算法、链式法则

实践案例: 用BP网络做样本数据分类

第18课 深度学习之CNN

卷积操作、Pooling操作

理解卷积网络、卷积网络训练以及GPU加速

卷积网络在图像分类中的应用

第19课 深度学习之RNN

RNN网络、LSTM

回归网络在自然语言处理中的应用: 字符模型和文本生成, char-rnn 案例

第20课 深度学习实践

Caffe 入门教程

Tensor Flow/MxNet 简介

实践案例: 使用Torch进行图像分类及卷积网络可视化的深度学习实践



2016春招公开课

我们在这里

7 七月算法 <http://www.julyedu.com/>

- 视频/课程/社区

- 七月题库APP: Android/iOS

- <http://www.julyapp.com/>

- 微博

- @研究者July

- @七月算法

- @邹博_机器学习

- 微信公众号

- julyedu



Any Question?

☐ 自由答疑



感谢大家！

恳请大家批评指正！

