**笔试题**

**姓名：（请填写姓名）**

**成绩：**

**时间：90分钟**

**笔试题说明：**

1. 所有题目必须从C、C++、Java、Scala四种语言中选择一种实现，否则不得分。
2. 请选择尽可能高效的算法，算法复杂度（大O表示法）比最优解高的话，得一半分。
3. 第5-7题，可任选2题作答

1. (10分)请将表示二进制整数的字符串“010110”转换成10进制整数。(如果你知道“二进制”的补码表示方式，请不要使用补码形式，这里的二进制是正常编码的二进制数)。不能使用已经存在的parseInteger()等库函数。算法复杂度O(n)

```C/C++

int binary\_to\_integer(char \*binary) {

}

```

```Java

int binary\_to\_integer(String binary) {

}

```

1. (10分)请实现如下函数, 计算并返回elems中最大的值和最小的值之差。算法复杂度O(n)

```C/C++

int diff(int \*elems, int size) {

}

```

```Java

int diff(int[] elems) {

}

```

1. (15分)如下定义了一个单向链表, 请实现一个函数将它反转.算法复杂度O(n)

```C/C++

struct Node {

int value;

struct Node *\*next;*

*};*

Node *\*reverse(Node \*head) {*

*}*

*```*

```Java

class Node {

public int value;

public Node next;

}

Node reverse(Node head) {

}

```

1. (15分)给定m个有序（递增）序列， 这些序列中包含的总的元素数目为n, 请将这些数组元素合并为一个有序（递增）序列。算法复杂度O(nm)

```C\C++

// 说明：sizes[ i]表示values[i]对应的数组的长度

int \*merge(int \*\*values, int \*sizes, int m) {

}

```

```Java

// 说明：seqs.size() == m

ArrayList<Integer> merge(ArrayList<ArrayList<Integer>> seqs) {

}

```

1. (25分)给定一个整数序列，存放在数组xs中，长度为n，请计算该序列的熵。

熵的计算方法：对于一个长度为n的序列xs，它包含m+1中不同的取值，s0, s1, …, sm，这些取值对应的出现概率分别是p0, p1, …,pm，则这个序列的熵为H(X) = -(p0\*log2(p0) + p1\*log2(p1) + … + pm\*log2(pm)). 其中，某个取值出现的概率p的计算方法为：这个取值出现的次数 除以 长度n。

```C\C++

double entropy(int \*xs, int n) {

}

```

```Java

double entropy(int[] xs) {

}

```

1. (25分)：在反欺诈中，每当一个账号发生一个行为的时候，都会对它进行一次风险判断，这次风险判断可能依赖本次行为以及本次行为之前的N-1次行为。其中N是一个常量。请实现一个数据结构，可以记录一个账号的最近N次行为的ID，并且可以根据下标读取这N次行为中的任何一个行为的ID。这个数据结构必须只能存有限个行为ID（一个账号的行为序列是无限的），并实现以下两个公共API，时间复杂度都必须是O（1）：

```C/C++

// 说明：对于C++，建议使用一个类来实现。

// 将当前事件id（event\_id）添加到数据结构中

void record(char \*event\_id);

// 获取下标为i的事件的事件id，最后一个事件的下标是0，保证i < N

char\* get\_last(int i);

```

```Java

class Records {

// 将当前事件id（event\_id）添加到数据结构中

public void record(String eventId);

// 获取下标为i的事件的事件id，最后一个事件的下标是0，保证 i < N

public String getLast(int i);

}

```

1. 附加题 (25分)给定一个二元组序列，[(s0, y0), (s1, y1), …]，该序列中的“s”和“y”分别存储在两个数组中ss和ys，两个数组的长度都是n，n>100。其中，对于任意合法的下标i，都有
   1. 0.0 < ss[i] < 1.0，代表一个模型评分
   2. ys[i]是0（负样本）或者1（正样本）

事实上，这个序列是一个模型在测试集上面的表现，ss[i]对应的是这个分类模型针对样本i输出的预测值，ys[i]是对应的是样本i的真值，请写代码计算该模型的KS值。

KS值计算方法：将测试集模型评分从低到高排序，在任意一个评分s上都可以计算：

cdf\_pos(s)：分数<=s的样本中，负样本数量占所有负样本数量的比例。cdf\_pos(s)：分数<=s的样本中，正样本数量占所有正样本数量的比例。

KS值就是上面这两个值的差值的最大值，代表了模型区分度的强弱。

KS = max{ cdf\_neg(s) – cdf\_pos(s) }

```C/C++

double kstest(double \*ss, int \*ys, int n) {

}

```

```Java

double kstest(double[] ss, int[] ys) {

}

```