

# 成都七中 2015 级奥赛学科夏令营

## 信息学能力测试

CDQZ OI Team

July 11, 2012

考试时间: 1 小时, 总分: 1023

题目说明: 由于考试时间十分有限, 题目的分值比较奇葩。一共 10 道题, 题目编号为 1-10。其中编号为  $i$  的题目分值为  $2^{(i-1)}$ 。请注意合理分配时间。

提示:  $\sum_{i=0}^{k-1} 2^i < 2^k$ ,

### 1. (Points for this problem: 1) 你好, 世界!

请在答题卡答题卡对应的横线上写下"Hello world!"(不含引号, 请仔细审题, 慎重作答)

### 2. (Points for this problem: 2) 看谁算得快

定义  $x! = 1 \times 2 \times 3 \cdots \times x$ 。

计算 25! 除以 1000 的余数。

### 3. (Points for this problem: 4) 0 与 1 的舞姿

对于二进制, 想必 NX 的你们早已有所了解。通俗的讲, 二进制就是逢 2 进位的进位制。十进制转化到二进制的方法也十分容易, 以 59 为例:

$$\begin{aligned} 59 \div 2 &= 29 \dots 1 \\ 29 \div 2 &= 14 \dots 1 \\ 14 \div 2 &= 7 \dots 0 \\ 7 \div 2 &= 3 \dots 1 \\ 3 \div 2 &= 1 \dots 1 \\ 1 \div 2 &= 0 \dots 1 \end{aligned}$$

那么倒着往上写则可得到  $(59)_{10} = (111011)_2$ 。

定义一个  $\mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z}$  的函数  $\oplus$ , 对于一位二进制数, 有  $1 \oplus 1 = 0$ ,  $1 \oplus 0 = 1$ ,  $0 \oplus 1 = 1$ ,  $0 \oplus 0 = 0$ 。对于高位二进制数, 可以独立每一位处理, 位数不等在前面补 0。

例如:  $(10110)_2 \oplus (101)_2 = (10011)_2$ 。

求:  $1 \oplus 2 \oplus 3 \oplus \dots \oplus 19 \oplus 20$  的值。

#### 4. (Points for this problem: 8) 圆与矩形

有一个  $m \times n$  的矩形, A 和 B 轮流向这个矩形内部放半径为  $r$  的圆。圆的每个部分都必须在这个矩形内, 并且圆与圆之间不能相交, 已经放好的圆不能移动。A 先开始轮流放, 谁不能继续放谁就输了。每场比赛给出  $m, n, r$ , 谁能赢得这场比赛? 如果 A 或 B 必胜, 则写出必胜方案, 否则说明不可能出现必胜。

#### 5. (Points for this problem: 16) 选择题 \*5

1. 本题的答案是: A. A; B. B; C. C; D. D;
2. 所有题里面所选选项最多的是: A. A; B. B; C. C; D. D;
3. 所有题目中所选选项恰好为 2 的是: A. A; B. B; C. C; D. D;
4. 奇数题号的题目中选 C 的有: A. 0 个; B. 1 个; C. 2 个; D. 3 个;
5. 本题答案和 X 题答案一样, X=: A. 1; B. 2; C. 3; D. 4;

#### 6. (Points for this problem: 32) 数的分解

对于每个  $\geq 2$  的正整数, 都可以将它分成若干个  $\geq 2$  的数的和。注意分解方案是有序的, 如  $5 = 2 + 3 = 3 + 2 = 5$ , 所以 5 有 3 种分解方案, 而 7 有 8 种。现在要你求 25 的分解方案数。

#### 7. (Points for this problem: 64) Transformer

如果说一个字母串是回文串, 那么就是说这个串从左往右读跟从右往左读是相同的。如 AYBBYA 是回文串, 而 ABBYA 不是。

现在给你一个的字母串, 你每次可以交换相邻两个字母, 你需要通过最少的交换次数, 使其变为回文串。

如将 ABBYY 变为回文串的最少次数为 4。

给定的字符串为: CCGGDBABEDCDDCDAED

#### 8. (Points for this problem: 128) 以毒攻毒

有 1000 瓶药, 其中有 999 瓶毒药, 1 瓶王老吉, 他们外观、重量都一模一样。现在要用一批小白鼠来找出王老吉。这个毒药有个特点: 每喝下一瓶毒药, 就会与之前喝下的毒药相中和, 使毒性消失。换句话说, 只有喝下奇数瓶毒药才会产生毒药。王老吉不会与毒药发生任何反应。现在你有一周的时间, 而毒药发作需要大约一周的时间。至少需要多少只小白鼠才能保证在一周内找出王老吉? 写出具体方案。(每瓶药可以使用多次)

#### 9. (Points for this problem: 256) The Last But One

令  $A$  是一个由非负整数组成的序列。我们想要把  $A$  中的每个数染成要么红要么蓝两个颜色, 满足一下要求:

- 至少一个元素被染成红色。
- 至少一个元素被染成蓝色。
- 所有红色数的 AND 的结果等于所有蓝色数 AND 的结果。

你得到了这个序列  $A$ , 你需要计算合法染色方案数。

定义一个  $\mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z}$  的函数 AND, 对于一位二进制数, 有  $1 \text{ AND } 1 = 1$ ,  $1 \text{ AND } 0 = 0$ ,  $0 \text{ AND } 1 = 0$ ,  $0 \text{ AND } 0 = 0$ 。对于高位二进制数, 可以独立每一位处理, 位数不等在前面补 0。

给定的序列  $A = \{13, 10, 4, 15, 4, 8, 4, 2, 4, 14, 12, 0\}$ 。

## 10. (Points for this problem: 512) 正则表达式

正则表达式是一个强大的字符串处理工具。在这里我们考虑其简化版本: egrep。

### 基本概念

字符包括大小写字母、数字、空格、以及各种符号。**字符串**是由字符链接产生的。例如 `a,b,+,` 都是字符, 而 `a+b_-c*d/e` 是字符串。一个字符串的**长度**只其包含的字符数。特别的, 一个字符也是一个长度为 1 的字符串。

**正则表达式 (Regular Expression)** 是一个能描述一个字符串集合的模式。设某个正则表达式为  $p$ , 它所能描述的字符串集合为  $A$ 。对于一个字符串  $s$ , 如果说  $p$  能**匹配** $s$ , 那么等价于  $s \in A$ 。

**连接 (Concatenation)** 是最基本的正则表达式操作。设有正则表达式  $p_1, p_2$ , 其对应的集合分别为  $A_1, A_2$ , 那么正则表达式  $p_1p_2$  对应的集合就是  $\{a_1a_2 | a_1 \in A_1, a_2 \in A_2\}$ 。

大多数字符本身就是一个正则表达式, 如字母、数字。它们能且仅能匹配自身。还有一些特殊的字符拥有特殊的含义, 我们称它们为**元字符**。

**点号 (Dot)**. 也是一个元字符, 它可以匹配任意字符。如果仅想匹配元字符本身, 需要在前面加转义符 `\`。例如, 可以用 `\.o\.` 来匹配 `.o.`。

**字符组 (Character Classes)**`[...]` 可以用来匹配若干字符之一。如想匹配所有单个数字, 则可以用 `[0123456789]`, 或者简写做 `[0-9]`。`[a-z]` 则表示所有小写字母。而 `[a-zA-Z]` 则匹配大小写所有字母。

如果我们想搜索单词“grey”, 但又不确定它是否写作“gray”, 就可以使用正则表达式 `gr[ae]y`。

类似的, 我们有排除性字符组 `[^...]`, 匹配不在其中的字符。如果要排除 `^` 可以用 `[^\^]`。

**行的起始与末尾 (Start and End of the Line)** 可以用元字符 `^` 和 `$` 来表示。`^` 匹配行首, `$` 匹配行尾。如 `^cat` 能匹配以下行:

```
cat
catch
category
```

类似的,我们可以用`\<`和`\>`分别匹配单词首和单词尾。这道题中,单词指连续一段由字母、数字构成的字符串。

**多选 (Alternation)|** 的意思是”或”。依赖它能够把不同的子表达式组合成一个总表达式。如`Alice|Bob`既能匹配`Alice`又能匹配`Bob`,但不能匹配`AliceBob`或者`Alice|Bob`。`|`的优先级非常低。如`6|7*`的含义与`6|(7*)`含义相同,不能匹配`677`。

**可选 (Option)?** 的意思是前面一项最多匹配一次,i.e. 可以匹配一次也可以不匹配。如`colou?r`可以匹配`color`与`colour`。可以用**括号(...)**来将一个子表达式变为可选。如`^[0-9]([0-9])?$`可以匹配一个一位数或者两位数。

除了可选元字符`?`,其他表示重复的还有:

- 星号`*` 前一项出现 0 次或多次。
- 加号`+` 前一项出现 1 次或多次。
- `{n}` 前一项恰好出现 `n` 次。
- `{n,}` 前一项出现 `n` 次以上。
- `{n,m}` 前一项出现次数在 `n` 次到 `m` 次之间。

比如要设计一个正则表达式来匹配身份证号,就可以用

`^[0-9]{15}|[0-9]{18}$`

**反向引用 (Backreferences)** 是括号的另一大用途。之前我们已经介绍过括号的两大用途:限制多选项的范围;将若干字符组合为一个单元,受问号或星号之类量词的作用。

在反向引用中,括号能”记住”他们包含的子表达式**匹配的文本**。例如,用`\<([0-9])\1\>`能匹配所有能被 11 整除的两位数。用`\n`能提取第 `n` 个括号的内容。如果使用多个括号,可以用`\1,\2,\3`等来表示第一、第二、第三组括号匹配的文本。括号是按照左括号(从左至右的出现顺序进行的)。

你也来当一回 *egrep!*

给出一个正则表达式,以及若干个字符串,写出所有能匹配到的字符串标号。(完全正确才有分)

(a) (Points for this problem: 32) subtask0

正则表达式:

`^[a-z][a-zA-Z0-9_]{4,15}$`

给出的字符串:

1. csimstu
2. zcwwzdjn
3. leocc
4. 844122492
5. jack950703
6. fater

7. rapid
8. Petr
9. nizhidaodetaiduole
10. tooyoung toosimple

(b) **(Points for this problem: 32)** subtask1

正则表达式:

`\<http://[-a-z0-9_.:]+/[-a-z0-9_:@&?+=, .!%$]*\.html?\>`

给出的字符串:

1. `http://www.google.com.hk`
2. `http://goojie/foo.html`
3. `http://www.goomei.com`
4. `http://www.goodi.html`

(c) **(Points for this problem: 64)** subtask2

正则表达式:

`^-[1-9][0-9]*\.[0-9]*|0\.[0-9]*[1-9]$`

给出的字符串:

1. `-.2`
2. `0.234`
3. `-1.0`
4. `-1.000`
5. `1.11`
6. `-5.012`
7. `.01`
8. `-9`
9. `1.100`
10. `1.1`
11. `1e10`

(d) **(Points for this problem: 128)** subtask3

正则表达式:

`^1$|^11+\1+$`

给出的字符串:

1. `1`
2. `11`
3. `111`
4. `1111`
5. `11111`
6. `111111`
7. `1111111`
8. `11111111`

9. 111111111  
10. 1111111111

*Let's Design!*

按照每道题目要求设计一个正则表达式来匹配给定集合  $A$  中的字符串,表达式的长度不能超过每题的限制。(Hint: 最好在开头和结尾加上 $^$ 与 $\$$ ) (“符合要求”一方面是指设计出来的正则表达式能匹配集合中的每一个字符串,另一方面集合以外的字符串都不能被匹配)

- (a) **(Points for this problem: 16)** subtask0  
 $A = \{\text{正则表达式自身}\}$ , 长度不超过 1。
- (b) **(Points for this problem: 16)** subtask1  
 $A = \{\text{美元金额(如果有小数点,则小数点后一定有 2 位)}\}$ , 长度不超过 30。
- (c) **(Points for this problem: 32)** subtask2  
 $A = \{\text{表示有意义时刻的字符串,例如 } 9:17 \text{ am}, 12:30 \text{ pm}, \text{但不能是 } 99:99 \text{ pm}\}$ 。长度不超过 40。
- (d) **(Points for this problem: 64)** subtask3  
 $A = \{1 \text{ 到 } 10000 \text{ 中 } 3 \text{ 的倍数}\}$ , 长度无限制。
- (e) **(Points for this problem: 128)** subtask4  
 $A = \{\text{长度为 } n \text{ 的字符串,使得 } 3x + 15y + 8z = n \text{ 存在非负整数解}\}$ , 长度不超过 40。