

华为OD机试 - 响应报文时间 (Java & JS & Python)

原创 伏城之外 已于 2023-06-03 01:27:21 修改 1239 收藏 5 版权
分类专栏: 华为OD机试AB (Java & JS & Python) 文章标签: 算法 华为机试 Java JavaScript Python

OD 华为OD机试AB (Ja... 同时被 2 个专栏收录
该专栏为热销专栏榜 第2名 ¥59.90 3382 订阅 371 篇文章 已订阅

题目描述

IGMP^Q 协议中，有一个字段称作最大响应时间 (Max Response Time)，HOST收到查询报文，解折出 MaxResponseTime 字段后，需要在 $(0, \text{MaxResponseTime}]$ 时间 (s) 内选取随机时间回应一个响应报文，如果在随机时间内收到一个新的查询报文，则会根据两者时间的大小，选取小的一方刷新回应时间。

最大响应时间有如下计算方式：

当 Max Resp Code < 128, Max Resp Time = Max Resp Code；

当 Max Resp Code ≥ 128 ,

0	1	2	3	4	5	6	7
+	+	+	+	+	+	+	+
	1		exp		mant		
+	+	+	+	+	+	+	+

Max Resp Time = (mant | 0x10) << (exp + 3);

注: exp最大响应时间的高5~7位: mant 为最大响应时间的低4位。

其中接收到的MaxRespCode 最大值为 255，以上出现所有字段均为无符号数。

现在我们认为 HOST收到查询报文时，选取的随机时间必定为最大值，现给出 HOST 收到查询报文个数 C，HOST 收到该报文的时间T，以及查询报文的最大响应时间字段值 M，请计算出HOST 发送响应报文的时间。

输入描述

第一行为查询报文个数 C，后续每行分别为 HOST 收到报文时间 T，及最大响应时间M，以空格分割。

输出描述

HOST 发送响应报文的时间。

备注

用例确定只会发送一个响应报文，不存在计时结束后依然收到查询报文的情况。

用例

输入	3 0 20 1 10 8 20
	输出 11
说明	收到3个报文， 第0秒收到第1个报文，响应时间为20秒，则要到0+20=20秒响应； 第1秒收到第2个报文，响应时间为10秒，则要到1+10=11秒响应，与上面的报文的响应时间比较获得响应时间最小为11秒； 第8秒收到第3个报文，响应时间为20秒，则要到8+20=28秒响应，与第上面的报文的响应时间比较获得响应时间最小为11秒； 最终得到最小响应报文时间为11秒

输入	2 0 255 200 60
	输出 260
说明	收到2个报文， 第0秒收到第1个报文，响应时间为255秒，则要到($15 0x10) << (7 + 3) = 31744$ 秒响应；(mant = 15, exp = 7) 第200秒收到第2个报文，响应时间为60秒，则要到200+60=260秒响应， 与第上面的报文的响应时间比较获得响应时间最小为260秒； 最终得到最小响应报文时间为260秒

题目解析

本题主要考察 [字符串操作](#)。

每一个报文都有一个最迟响应时间，即 $t + \text{getMaxResponseTime}(m)$ ，我们只需要求出最小的最迟响应时间即可。

关于 `getMaxResponseTime` 的逻辑：

当 $m < 128$ 时， m 就是 `MaxResponseTime`

当 $m \geq 128$ 时，需要将 m 转化为八位二进制字符串，取 1~3 范围作为 `exp`，取 4~7 范围作为 `mant`

然后再将 `exp` 和 `mant` 从二进制串转化为十进制，带入公式： $(mant | 16) << (exp + 3)$ 求解即可。

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         int c = sc.nextInt();
8
9         int[][] messages = new int[c][2];
10        for (int i = 0; i < c; i++) {
11            messages[i][0] = sc.nextInt();
12            messages[i][1] = sc.nextInt();
```

```
13     }
14
15     System.out.println(getResult(messages));
16 }
17
18 public static int getResult(int[][][] messages) {
19     int ans = Integer.MAX_VALUE;
20
21     for (int[] message : messages) {
22         int respTime = message[0] + getMaxResponseTime(message[1]);
23         ans = Math.min(ans, respTime);
24     }
25
26     return ans;
27 }
28
29 public static int getMaxResponseTime(int m) {
30     if (m >= 128) {
31         StringBuilder bStr = new StringBuilder(Integer.toBinaryString(m));
32
33         while (bStr.length() < 8) {
34             bStr.insert(0, '0');
35         }
36
37         int exp = Integer.parseInt(bStr.substring(1, 4), 2);
38         int mant = Integer.parseInt(bStr.substring(4), 2);
39
40         return (mant | 16) << (exp + 3);
41     } else {
42         return m;
43     }
44 }
45 }
```