# 腾讯极客技术挑战赛-第三期：码上种树 1-200W writeup

原创

kang0x0 于 2021-03-18 11:17:50 发布 3121 收藏 3

分类专栏： 腾讯极客技术挑战赛 文章标签： js python



本文链接：https://blog.csdn.net/kang0x0/article/details/114946505




腾讯极客技术挑战赛 专栏收录该内容

2 篇文章 0 订阅

订阅专栏

## 腾讯极客技术挑战赛-第三期：码上种树 1-200W writeup

### 0x00 简要说明

- 赛题链接：http://geek.qq.com/tree/#
- 因为不会JS，所以不知道更好的分析方法，这里仅提供一种解题方法。但后面的题目还是需要看一下JS的基础教程的。
- 主要的解题过程是：通过浏览器F12的功能，调试输出代码的运算过程和结果，猜测计算a的算法，然后用Python实现，测试验证修改。

### 0x01 签到

- 第一题直接查看数据包，发现第一次响应包返回的数据中的参数a的值，就是第二次请求发送参数a的值

```
# 第一题
import requests
while True:
 # 第一次请求url
 url = "http://159.75.70.9:8081/pull?u="+token
 # 发送get请求
 r = requests.get(url)
 # 获取返回的json数据
 print(r.json())
 t = r.json()['t']
 a = r.json()['a']
 c = r.json()['c']
 # 第二次请求url
 url = "http://159.75.70.9:8081/push?t=" + t + "&a="+ str(a[0])
 # 发送get请求
 r = requests.get(url)
 # 获取返回的json数据
 print(r.json())
```

---

### 0x02 简单计算

- 第二题没有记录题目，因为不熟悉JS，所以在浏览器F12下查看相关文件，发现参数a经过运算后返回（具体是哪里看到的，忘了）

```python
# 第二题
import requests
while True:
 # 第一次请求url
 url = "http://159.75.70.9:8081/pull?u="+token
 # 发送get请求
 r = requests.get(url)
 # 获取返回的json数据
 print(r.json())
 t = r.json()['t']
 a = r.json()['a']
 c = r.json()['c']
 # 第二次请求url
 url = "http://159.75.70.9:8081/push?t=" + t + "&a="+ str(a[0]*a[0]+a[0])
 # 发送get请求
 r = requests.get(url)
 # 获取返回的json数据
 print(r.json())
```

---

## 0x03 变量名替换

- 第三题 发现题目如下，思路：主要对开头为“_”的变量进行替换，方便阅读，注意不要把 数字替换掉

```
//第三题 题目
var _0xe936=['A5473788'];
(function(_0x48e85c,_0xe936d8)
{var _0x23fc5a=function(_0x2858d9)
{while(--_0x2858d9)
    {_0x48e85c['push'](_0x48e85c['shift']());}};
    _0x23fc5a(++_0xe936d8);}(_0xe936,0x196));
    var _0x23fc=function(_0x48e85c,_0xe936d8)
    {_0x48e85c=_0x48e85c-0x0;var _0x23fc5a=_0xe936[_0x48e85c];
    return _0x23fc5a;};
    window[_0x23fc('0x0')]=function(_0x335437)
    {var _0x1aac02=0x30d3f;
    for(var _0x3bed6a=0x30d3f;_0x3bed6a>0x0;_0x3bed6a--)
    {var _0x375340=0x0;
    for(var _0x1ddb77=0x0;_0x1ddb77<_0x3bed6a;_0x1ddb77++)
    {_0x375340+=_0x335437['a'][0x0];}
    _0x375340%_0x335437['a'][0x2]==_0x335437['a'][0x1]&&_0x3bed6a<_0x1aac02&&(_0x1aac02=_0x3bed6a);}
    return _0x1aac02;};

//替换变量名后如下，然后分析计算过程，发现用到参数a的地方，最后python实现运算过程
var _0xe936=['A5473788'];
(function(x,y)
{
    var result=function(z)
    {
        while(--z)
        {
            x['push'](x['shift']());
        }
    };
    result(++y);
}
    (_0xe936,0x196)
);
    var _0x23fc=function(x,y)
    {
        x=x-0x0;
        var result=_0xe936[x];
        return result;
    };

    window[_0x23fc('0x0')]=function(x1)
    {
    var x2=0x30d3f;
    for(var i=0x30d3f;i>0x0;i--)
    {
        var j=0x0;
        for(var k=0x0;k<i;k++)
        {
            j+=x1['a'][0x0];
        }
        j%x1['a'][0x2] == x1['a'][0x1] && i<x2 && (x2=i);
    }
    return x2;
};
```

```python
# 第三题 解
import requests
while True:
 # 第一次请求url
 url = "http://159.75.70.9:8081/pull?u="+token
 # 发送get请求
 r = requests.get(url)
 # 获取返回的json数据
 print(r.json())
 t = r.json()['t']
 a = r.json()['a']
 c = r.json()['c']
 # 经过变量名替换，发现运算过程如下：
 x = 0x30d3f
    for i in range(0x30d3f, 0, -1):
        j = a[0x0] * i
        if (j % a[0x2] == a[0x1] and i < x):
            x = i
    print(x)

 # 第二次请求url
 url = "http://159.75.70.9:8081/push?t=" + t + "&a="+ str(x)
 # 发送get请求
 r = requests.get(url)
 # 获取返回的json数据
 print(r.json())
```

---

## 0x04 JS + 异步编程

- 第四题 原本的题目没有记录。
- 主要是需要将JSFuck（中括号之间的内容）解密成函数名，并替换; 相关的参数也替换一下，得到如下代码；
  分析代码，发现用到了JS异步编程的内容（需要自行了解相关内容）；知道大概的功能后，猜测代码的主要作用是循环把参数a的值 执行 + 、- 、* 的运算；最后用 Python 实现 运算过程求参数a。
- 需要注意的地方：
  1、通过F12控制台输出运算的过程，发现 参数a 的值 已经排好序，再遍历执行；
  2、x2是先++，所以第一次是从 减法 运算；
  3、最后结果 取反。

```
//第四题 题目
window.A593C8B8
=async(text)=>
(
    (window,text,x2,x3,x4)=>
    {
        let x5=function*()
        {
            while([])yield[(_, __)=> _+__, (_, __)=> _-__, (_, __)=> _ * __][++x2 % 3].bind(null, x3, x4)
        }();

        let f6=function(x5,x6,x7)
        {
            x4=x5;
            x3=x6.next().value();
            x2==text.a.length && x7(-x3)
        };

        return new
        Promise(
            x2=>text.a.forEach( x3=>window.setTimeout(x4=>f6(x3,x5,x2), x3))
        )
    }
)(window,text,0,0,0)
```

```python
# 第四题 解
import requests

def add(x, y):
    return x + y

def sub(x, y):
    return x - y

def mul(x, y):
    return x * y

fun = [add, sub, mul]

while True:
 # 第一次请求url
 url = "http://159.75.70.9:8081/pull?u="+token
 # 发送get请求
 r = requests.get(url)
 # 获取返回的json数据
 print(r.json())
 t = r.json()['t']
 a = r.json()['a']
 c = r.json()['c']
 # 第四关 求参数a 的运算过程
 answer = 0
    i = 0
    a.sort()
    for num in a:
        i += 1
        answer = fun[i % 3](answer, num)
    print(-answer)

 # 第二次请求url
 url = "http://159.75.70.9:8081/push?t=" + t + "&a=" + str(-answer)
 # 发送get请求
 r = requests.get(url)
 # 获取返回的json数据
 print(r.json())
```

---

## 0x05 WebAssembly

- 第五题 发现JS使用了WebAssembly这个东西，具体的内容自行百度。
- 参考下方“理解WebAssembly文本格式”链接，将运算过程用 Python 实现。
- 题目代码主要实现：找出参数a[0]中每一位的 最大和最小 值，然后相乘，
  并加上上次的执行结果，共执行a[1]次（可优化）。
- 参考链接：理解WebAssembly文本格式

```
//第五题 题目
(module
  (func $import0 (import "Math" "min") (param i32 i32) (result i32))
  (func $import1 (import "Math" "max") (param i32 i32) (result i32))
  (export "Run" (func $func2))
  (func $func2 (param $var0 i32) (param $var1 i32) (result i32)
    (local $var2 i32) (local $var3 i32) (local $var4 i32) (local $var5 i32) (local $var6 i32) (local $var7 i32)
    local.get $var0
    local.set $var2
    local.get $var1
    i32.const 1
    i32.sub
    local.tee $var4
    if
      loop $label1
        local.get $var2
        local.set $var3
        i32.const 0
        local.set $var6
        i32.const 10
        local.set $var7
        loop $label0
          local.get $var3
          i32.const 10
          i32.rem_u
          local.set $var5
          local.get $var3
          i32.const 10
          i32.div_u
          local.set $var3
          local.get $var5
          local.get $var6
          call $import1
          local.set $var6
          local.get $var5
          local.get $var7
          call $import0
          local.set $var7
          local.get $var3
          i32.const 0
          i32.gt_u
          br_if $label0
        end $label0
        local.get $var2
        local.get $var6
        local.get $var7
        i32.mul
        i32.add
        local.set $var2
        local.get $var4
        i32.const 1
        i32.sub
        local.tee $var4
        br_if $label1
      end $label1
    end
    local.get $var2
  )
)
```

- Python主要实现如下功能
  1、通过将参数a[0],转换为字符串，遍历字符串 找 最大值 和 最小值
  2、经过测试发现 计算运算 多次 之后，结果将保持不变；用temp保存上次的结果，并比较，如果相等则退出循环，不需要执行a[1]次。

```
# 第五题 解
import requests

answer = 0
i = 0
temp = 0

def func3(var0, var1):
    var2 = var0
    var4 = var1 - 1
    temp = var2
    if var4 != 0:
        while True:
            var3 = var2
            var6 = 0
            var7 = 10
            # print(var3)
            for i in str(var3):
                if int(i) > var6:
                    var6 = int(i)
                if int(i) < var7:
                    var7 = int(i)
            var2 = var6 * var7 + var2
            if (var2 == temp):
                return var2
            temp = var2
            var4 = var4 - 1
            if var4 == 0:
                break
    return var2

while True:
 # 第一次请求url
 url = "http://159.75.70.9:8081/pull?u="+token
 # 发送get请求
 r = requests.get(url)
 # 获取返回的json数据
 print(r.json())
 t = r.json()['t']
 a = r.json()['a']
 c = r.json()['c']

 # 第五关 求参数a 的运算过程
    answer = func3(a[0], a[1])
    print(answer)

 # 第二次请求url
 url = "http://159.75.70.9:8081/push?t=" + t + "&a=" + str(answer)
 # 发送get请求
 r = requests.get(url)
 # 获取返回的json数据
 print(r.json())
```

## 0x06 JS_VM

- 说明：自己写的代码有点烂，没有把获取e的值写出函数形式，其他地方可能也写的不够好。
- 第六关 题目每1W JS会变，但是算法不变。简要分析过程如下
  1、如何找到运算过程：不会JS，但发现可以用console.log(x)输出内容，于是将运算过程都输出到控制台分析；
  2、经过分析后得知 程序 会从固定的位置 获取 “种子”值 x1；然后乘上上一次的运算结果，在固定位置获取 “求余”值 p，对运算结果求余；然后比较相乘的次数是否大于a[i]，如果大于，则从另一个位置 获取“种子”值x2，重复上述过程；
  3、通过修改参数a的值为1，分析后续的运算过程；可以得知，算法的过程如下：
  (x1^a[0] * x2^a[1] + x3^a[3] * x4^a[4] + … + x11^a[10] * x12^a[11]) % p
- 解决的方法：快速幂（感谢大佬的提示）
- 最后通过Python脚本 实现自动化获取 数据

```
# 第六关 解
# 获取js中参数e的数据
e = []
c = "F3B9B832"
url = "http://159.75.70.9:8080/" + c + ".js"
print(url)
# 发送get请求
r = requests.get(url)
# 获取返回的json数据
start = r.text.find("__TENCENT_CHAOS_VM(0,[") + len("__TENCENT_CHAOS_VM(0,[")
end = r.text.find("],window)}()")
strList = r.text[start: end].split(',')
for take_str in strList:
    e.append(int(take_str))
print("e:", len(e))

# 从 e 中 获取 求余值 和 种子值 列表
judge = e[142]
seedList = []
seedStart = 142
step = [149, 239]
index = seedStart;
# 求余值
modStart = 2361
mod = 0
s = ''
i = modStart
while i < len(e):
    if e[i] == judge:
        s += str(e[i + 1] - 48)
        i += 1
    else :
        mod = int(s)
        break
    i += 1
print(mod)

# 种子值
count = 0;
while True:
    s = ''
    if count >= 12:
        break
    i = index
```

```
    while i < len(e):
        if e[i] == judge:
            s += str(e[i + 1] - 48)
            i += 1
        else:
            seedList.append(int(s))
            break
        i += 1
    index += step[count%2]
    count += 1
print(seedList)

# 快速幂
def fastPower(ans, a, b, n):
    while n > 0:
        if n % 2 == 1:
            ans = ans * a % b
        a = a * a % b
        n = n // 2
    return ans % b

count = 0
while(count < 10000):
    print(count)
    # 请求地址
    url = "http://159.75.70.9:8081/pull?u="+token
    # 发送get请求
    r = requests.get(url)
    # 获取返回的json数据
    print(r.json())
    t = r.json()['t']
    a = r.json()['a']
    c = r.json()['c']

    # 第六关
    ans = 1
    addNumber = 0
    i = 0
    while i < 6:
        j = 0
        while j < 2:
            ans = fastPower(ans, seedList[i*2 + j], mod, a[i*2 + j])
            j += 1
        addNumber = (addNumber + ans) % mod
        ans = 1
        i += 1
    # print(addNumber)
    url = "http://159.75.70.9:8081/push?t=" + t + "&a=" + str(addNumber)
    print(url)

 # 发送get请求
    r = requests.get(url)
    # 获取返回的json数据
    print(r.json())
    try:
        if r.json()["error"] != None:
            url = "http://159.75.70.9:8080/" + c + ".js"
            print(url)
            # 发送get请求
            r = requests.get(url)
```

```
            # 获取返回的json数据
            start = r.text.find("__TENCENT_CHAOS_VM(0,[") + len("__TENCENT_CHAOS_VM(0,[")
            end = r.text.find("],window)}()")
            strList = r.text[start: end].split(',')
            e.clear()
            for take_str in strList:
                e.append(int(take_str))
            print("e:", e)

            judge = e[142]
            print(judge)
            # 获取种子
            i = modStart
            while i < len(e):
                if e[i] == judge:
                    s += str(e[i + 1] - 48)
                    i += 1
                else:
                    mod = int(s)
                    break
                i += 1
            print(mod)

            # 种子
            j = 0; index = seedStart;
            seedList.clear()
            while True:
                s = ''
                if j >= 12:
                    break
                i = index
                while index < len(e):
                    if e[i] == judge:
                        s += str(e[i + 1] - 48)
                        i += 1
                    else:
                        seedList.append(int(s))
                        break
                    i += 1
                index += step[j % 2]
                j += 1
            print(seedList)
            count += 1
    except KeyError:
        count += 1
        continue
```

## 0x07 200W+1

- 说明：最后两棵树请看大神的writeup。