



中国计算机学会
China Computer Federation



如何使用 tuack 命题

清华大学 杨耀良



中国计算机学会
China Computer Federation



命题中的需求？

- 如何生成格式化的题面？
 - word、latex？
- 如何管理数据？
 - 子任务？
- 如何对数据进行测试？
 - 命题人？ 验题人？
- 如何对比赛进行管理？
 - 比赛题目配置？ 加题、删题？



中国计算机学会
China Computer Federation



tuack 是什么

- tuack 是一个开源命题工具
 - 规范了题面的书写方式，生成格式化的题面
 - 方便了对比赛的管理
 - 方便对数据的管理和测试
 - 需要会一些命令行操作
 - 基于 python，可以直接用 pip 安装
 - wiki 地址: https://gitee.com/mulab/oi_tools/wikis/home
 - 代码仓库: https://gitee.com/mulab/oi_tools



中国计算机学会
China Computer Federation



tuack 的安装

- python 环境安装
 - 推荐使用 anaconda
 - <https://mirrors.bfsu.edu.cn/anaconda/archive/>
- 使用 pip 安装 tuack
 - pip install tuack



中国计算机学会
China Computer Federation



tuack 的安装

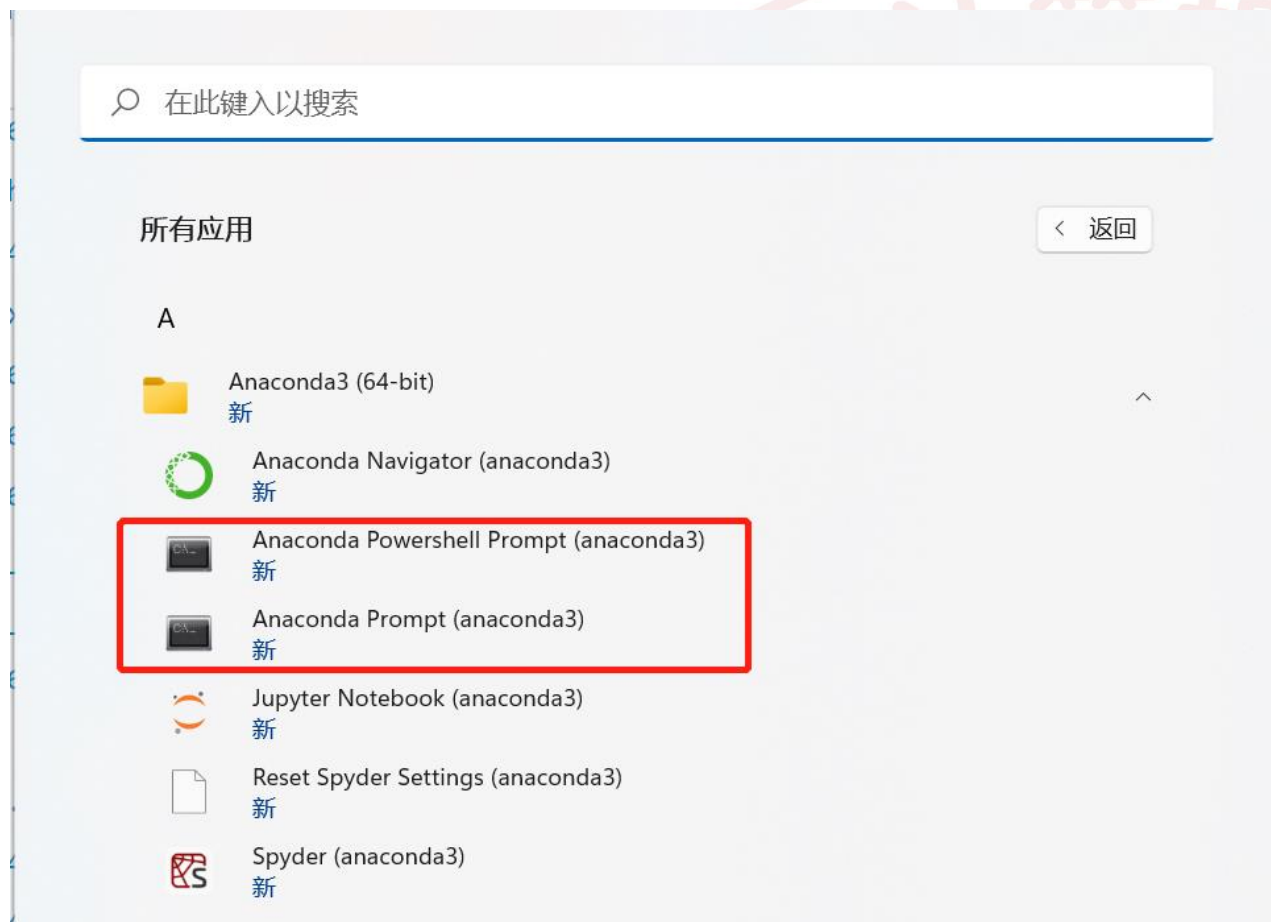
Anaconda3-2022.05-Windows-x86_64.exe	593.9 MiB	2022-05-11 02:36
Anaconda3-2022.10-Linux-aarch64.sh	534.5 MiB	2022-10-18 05:24
Anaconda3-2022.10-Linux-ppc64le.sh	360.0 MiB	2022-10-18 05:24
Anaconda3-2022.10-Linux-s390x.sh	282.4 MiB	2022-10-18 05:24
Anaconda3-2022.10-Linux-x86_64.sh	737.6 MiB	2022-10-18 05:24
Anaconda3-2022.10-MacOSX-arm64.pkg	484.1 MiB	2022-10-18 05:24
Anaconda3-2022.10-MacOSX-arm64.sh	472.5 MiB	2022-10-18 05:25
Anaconda3-2022.10-MacOSX-x86_64.pkg	688.6 MiB	2022-10-18 05:25
Anaconda3-2022.10-MacOSX-x86_64.sh	681.6 MiB	2022-10-18 05:25
Anaconda3-2022.10-Windows-x86_64.exe	621.2 MiB	2022-10-18 05:26
Anaconda3-4.0.0-Linux-x86.sh	336.9 MiB	2017-01-31 01:34
Anaconda3-4.0.0-Linux-x86_64.sh	398.4 MiB	2017-01-31 01:35
Anaconda3-4.0.0-MacOSX-x86_64.pkg	341.5 MiB	2017-01-31 01:35
Anaconda3-4.0.0-MacOSX-x86_64.sh	292.7 MiB	2017-01-31 01:36
Anaconda3-4.0.0-Windows-x86.exe	283.1 MiB	2017-01-31 01:36
Anaconda3-4.0.0-Windows-x86_64.exe	345.4 MiB	2017-01-31 01:37
Anaconda3-4.1.0-Linux-x86.sh	328.4 MiB	2017-01-31 01:38



中国计算机学会
China Computer Federation



tuack 的安装

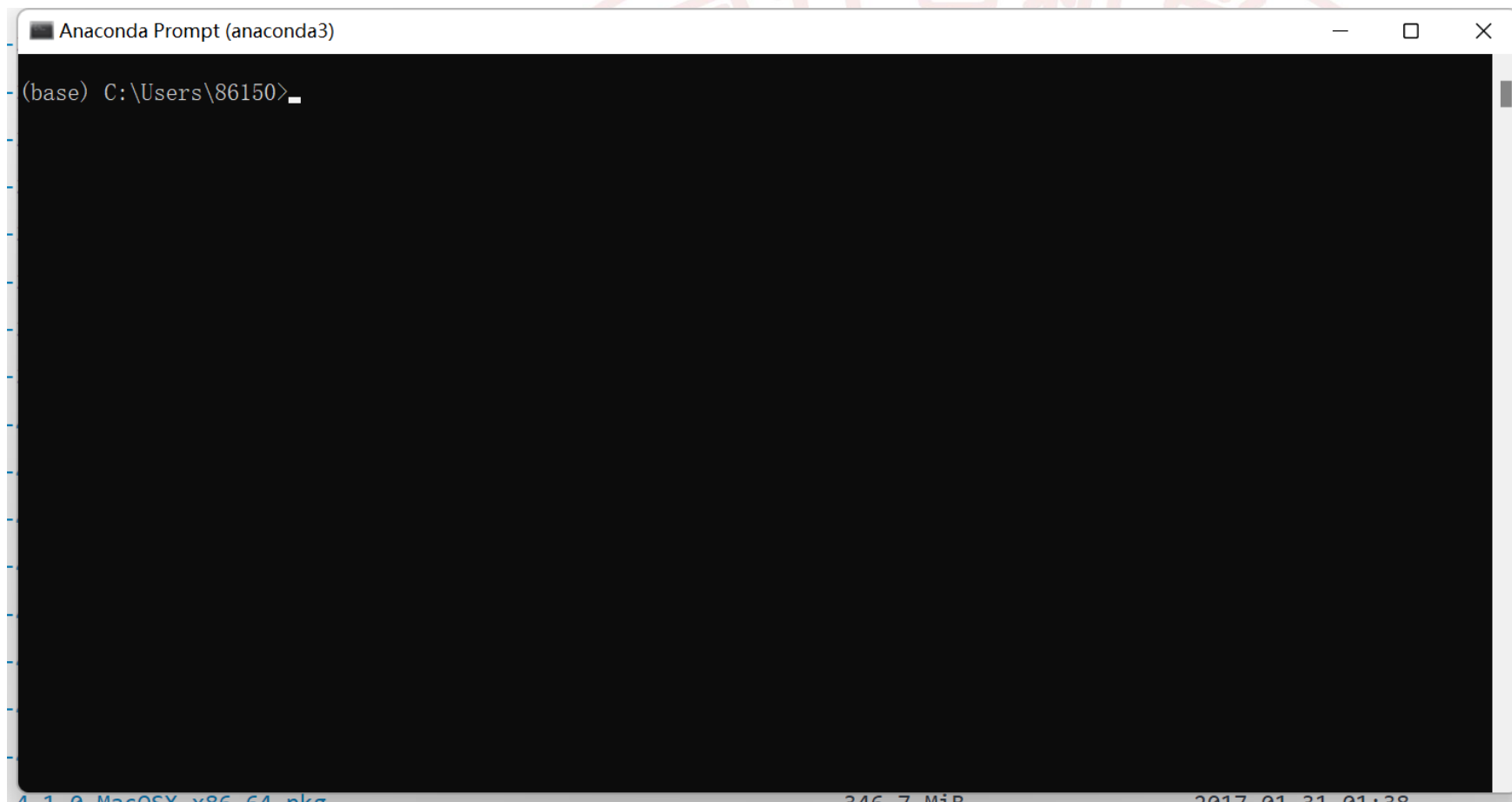




中国计算机学会
China Computer Federation



tuack 的安装





中国计算机学会
China Computer Federation



tuack 的安装

```
Anaconda Prompt (anaconda3)

(base) C:\Users\86150>cd Desktop

(base) C:\Users\86150\Desktop>mkdir test

(base) C:\Users\86150\Desktop>cd test

(base) C:\Users\86150\Desktop\test>python -m tuack.gen contest
C:\Users\86150\anaconda3\python.exe: Error while finding module specification for 'tuack.gen' (ModuleNotFoundError: No module named 'tuack')

(base) C:\Users\86150\Desktop\test>pip install tuack
Collecting tuack
  Downloading tuack-0.1.4.12.6.tar.gz (1.8 MB)
    ----- 1.8/1.8 MB 1.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting jinja2>=3.0
  Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
Collecting natsort>=6.0.0
  Using cached natsort-8.2.0-py3-none-any.whl (37 kB)
Requirement already satisfied: pyyaml>=5.1 in c:\users\86150\anaconda3\lib\site-packages (from tuack) (6.0)
Collecting rarfile>=3.0
  Using cached rarfile-4.0-py3-none-any.whl (28 kB)
Collecting requests_toolbelt>=0.9.1
  Using cached requests_toolbelt-0.10.1-py2.py3-none-any.whl (54 kB)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\86150\anaconda3\lib\site-packages (from jinja2>=3.0->tuack) (2.0.1)
Requirement already satisfied: requests<3.0.0,>=2.0.1 in c:\users\86150\anaconda3\lib\site-packages (from requests_toolbelt>=0.9.1->tuack) (2.28.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\86150\anaconda3\lib\site-packages (from requests<3.0
```




如何使用 tuack 创建比赛

- 首先在命令行中切换至准备好的文件夹
- 使用 `python -m tuack.gen contest` 命令创建比赛
- 使用 `python -m tuack.gen day day0 [day1] ... [dayn]` 命令创建比赛日
- 进入比赛日的文件夹
- 使用 `python -m tuack.gen problem A B C` 命令创建三个分别名为 A B C 的题目模板



中国计算机学会
China Computer Federation



如何使用 tuack 配置比赛题目

- 可以将其他题目的文件夹放置在比赛日文件夹内
- 直接修改比赛日文件夹内的 `conf.yaml` 文件配置该比赛日的题目



tuack 题目文件夹结构

- https://gitee.com/mulab/oi_tools/wikis/%E6%96%87%E4%BB%B6%E7%9A%84%E5%AD%98%E6%94%BE%E5%92%8C%E5%AE%9A%E4%B9%89

名称	修改日期	类型	大小
data	2023/1/3 13:23	文件夹	
down	2023/1/3 13:23	文件夹	
pre	2023/1/3 13:23	文件夹	
resources	2023/1/3 13:23	文件夹	
solution	2023/1/3 13:23	文件夹	
statement	2023/1/3 13:23	文件夹	
tables	2023/1/3 13:23	文件夹	
.gitignore	2023/1/3 13:23	Git Ignore 源文件	1 KB
conf.yaml	2023/1/3 13:23	Yaml 源文件	1 KB



如何使用 tuack 写题面

- https://gitee.com/mulab/oi_tools/wikis/%E9%A2%98%E9%9D%A2%E7%9A%84%E4%B9%A6%E5%86%99
- 编辑 statement 文件夹下的 zh-cn.md
- 英文题面需要新建 en.md 文件
- 几个例子
- 如何渲染 pdf 题，需要安装 XeLaTeX：
https://gitee.com/mulab/oi_tools/wikis/%E7%9B%B8%E5%85%B3%E5%B7%A5%E5%85%B7
- 使用 `python -m tuack.ren noi -o Linux` 指令渲染题面



中国计算机学会
China Computer Federation



简单的题面例子

```
{{ self.title() }}
```

```
{{ s('description') }}
```

给定一个正整数 k ，有 k 次询问，每次给定三个正整数 n_i, e_i, d_i ，求两个正整数 p_i, q_i ，使 $n_i = p_i \times q_i, e_i \times d_i = (p_i - 1)(q_i - 1) + 1$ 。

```
{{ s('input format') }}
```

```
{{ self.input_file() }}
```

第一行一个正整数 k ，表示有 k 次询问。

接下来 k 行，第 i 行三个正整数 n_i, d_i, e_i 。

```
{{ s('output format') }}
```

```
{{ self.output_file() }}
```

输出 k 行，每行两个正整数 p_i, q_i 表示答案。

为使输出统一，你应当保证 $p_i \leq q_i$ 。

如果无解，请输出 `NO`。

```
{{ s('sample', 1) }}
```

```
{{ self.sample_text() }}
```

```
{{ s('sample', 2) }}
```

```
{{ self.sample_file() }}
```

```
{{ s('sample', 3) }}
```

```
{{ self.sample_file() }}
```

```
{{ s('sample', 4) }}
```

```
{{ self.sample_file() }}
```

```
{{ s('数据范围') }}
```

以下记 $m = n - e \times d + 2$ 。

保证对于 100% 的数据， $1 \leq k \leq 10^5$ ，对于任意的 $1 \leq i \leq k$ ， $1 \leq n_i \leq 10^{18}, 1 \leq e_i \times d_i \leq 10^{18}, 1 \leq m \leq 10^9$ 。

```
{{ tbl('data') }}
```



数据表格例子

- 直接使用 data.pyinc 渲染
- 修改 data.pyinc:

```
1 ret = [["测试点编号", " $n,m,q \\leq$ ", "特殊条件"]]
2 ret.append([" $1$ ", " $200$ ", " $1, 2$ "])
day0 > p0 > tables > {} table.json > ...
1 [
2     ["Test cases", "$n$", "$a_i$"],
3     ["1-3", "$<100$", "$<100000$"],
4     ["4-7", "$\\le 3000$", "$<100$"],
5     ["8-10", "$\\le 3000$", "$<100000$"]
6 ]
7 [
8     ["18 ~ 20", "$10^5$", "无"]
9 ]
10 ret.append([" $10 \\sim 17$ ", " $10^5$ ", " $2$ "])
11 ret.append([" $18 \\sim 20$ ", " $10^5$ ", "无"])
12 return merge_ver(ret)
13
14
15
```

测试点编号	$n, m, q \leq$	特殊条件
1	200	1, 2
2		1
3		2

Test cases	n	a_i
1-3	< 100	< 100000
4-7	< 1000	< 100
8-10	< 1000	< 100000
18 ~ 20		无



中国计算机学会
China Computer Federation



如何使用 tuack 进行测试

- 编辑 conf.yaml 文件配置题目
 - 修改 data 和 user 字段
- 使用 `python -m tuack.test` 指令进行测试
- 几个例子



中国计算机学会
China Computer Federation



题目配置文件例子

```
! conf.yaml •
CSP-S > game > ! conf.yaml
1  args:
2    n: 100000
3    m: 100000
4    q: 100000
5  compile:
6    c: -O2 -std=c11
7    cpp: -O2 -std=c++14
8    java: ''
9    pas: -O2
10   py: ''
11  data:
12    - cases: [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
13  folder: problem
14  memory limit: 512 MiB
15  name: game
16  partial score: false
17  pre: []
18  samples:
19    - cases: [1, 2, 3, 4]
20  time limit: 1.0
21  title:
22    zh-cn: 策略游戏
23  type: program
24  users:
25    A:
26      std: {expected: == 100, path: A/std.cpp}
27    B:
28      bf: {expected: == 60, path: B/bf.cpp}
29      std: {expected: == 100, path: B/std.cpp}
30  version: 2
31
```



中国计算机学会
China Computer Federation

QA



• 多人协作？使用 Git 进行同步

- git add
- git commit
- git pull
- git push





中国计算机学会
China Computer Federation



谢谢大家，欢迎提问