

Đặc tả project bấm giờ

December 23, 2020

1 Nhu cầu

Luyện tập CP, rõ hơn là từ đề bài toán tới một đoạn mã giải quyết trên máy tính cần trải qua nhiều bước. Giả sử là requirement (yêu cầu) là rõ ràng và không yêu cầu chuẩn hóa đoạn mã đầu ra, cũng phải trải qua nhiều giai đoạn: đọc đề, hiểu đề và theo dấu các mẫu input / output đề cho, vừa tạo thêm vài input/output khác để hiểu rõ hơn vừa quan sát các tính chất nếu có, đơn giản hóa bài toán bằng cách sắp xếp lại/thêm giả thiết, chứng minh một số tính chất, đưa ra thuật toán, code, test, debug, mỗi giai đoạn có thể lặp lại nhiều lần. Để cải thiện tốc độ làm việc, ta cần phải biết "nút thắt cổ chai" của quá trình giải nằm ở giai đoạn nào.

Muốn làm được điều đó, Mostafa Saad Ibrahim dạy rằng ta cần phải *logging* mỗi lần ngồi vô máy giải bài. Mỗi giờ làm cái gì, ghi chú lại. Khi kết thúc (AC, WA bỏ bài, bận...) thì ngồi lại phân tích có gì chưa tốt không. Vậy đang cần một ứng dụng.

2 Yêu cầu

- Thời gian lập trình phải nhanh. Tệ nhất là mất 2-3 ngày.
- Không cần giao diện bắt mắt, gõ lệnh trên console là đủ. Thời gian khởi động phải nhanh.
- Sau khi kết thúc làm việc, tự dump ra file log các mốc thời gian tới phút, lượng thời gian đã mất.

3 Đặc tả

- Dùng ngôn ngữ lập trình Python, khởi động chương trình bằng terminal.
- Sau khi khởi động, người dùng gõ lệnh vào terminal để điều khiển chương trình. (gõ lệnh và kết thúc bằng phím Enter)
- Các lệnh có thể được nhập được thể hiện trong bảng sau:

Câu lệnh	Biểu thị	Ví dụ
na <Tên phiên làm việc> go done exit	Đặt tên phiên làm việc Bắt đầu phiên làm việc Kết thúc phiên làm việc (sẵn sàng lưu log) Lưu log ra file và thoát chương trình	na 875A, na #375d1 go done exit
rd st [<Mô tả>] ob [<Nhận xét>] al [<Mô tả ngắn gọn thuật>] im [<Mô tả>] db [<Mô tả bug>] gg [<Nani>]	Đọc đề Đọc sample test trace sample test và tạo ra 1, 2 test tay Nhận xét (chương trình tự đánh số) nên viết ob bằng ANSI trên terminal Đề ra thuật (ct đánh số) Bắt đầu cài thuật Debug Google thứ gì đó Dấu hiệu mất tập trung / gian lận	rd st, st 2 custom test ob, ob a!=b al bfs+dsu im py, im c++ db sample 2, db rte gg wheel fctorizatn, gg MTP gg hen
wa [<Mô tả>] tle [<Mô tả>] mle [<Mô tả>] rte [<Mô tả>] ac [<Mô tả>] fu [<Mô tả>] tc [<Mô tả>] afk [<Mô tả>] bck [<Mô tả>] ed [<Mô tả>]	Wrong answer Time limit exceeded Memory limit exceeded Run time error Accepted FUCK UP Mở test case đáp án (vd Codeforces) Rời khỏi máy tính Quay lại máy tính Đọc editorial	wa 13 tle 7 mle 5 rte 6 ac, ac py fuuuuuuuuuuuuu tc 14 afk chore bck ed sol, ed extended euclid

- Một phiên làm việc điển hình bắt đầu bằng việc tùy hứng là đặt tên, <go>, làm các thứ gì đó, <done>, <exit> để kết thúc chương trình.
- Nên nhập lệnh trước khi làm (sau khi làm cũng được, tùy...)
- Nếu có một phiên làm việc đang diễn ra, lệnh <done> chưa được nhập mà lệnh <exit> đã được nhập thì phiên hiện tại cũng được lưu.
- Sau khi chương trình kết thúc, log được tự động ghi ra file, tên file gồm ngày giờ và tên phiên làm việc.

Hết rồi, đặc tả nhiều đó thôi, cài đặt đi thẳng ngu.

4 Ví dụ

Để cho vài ví dụ chạy chương trình. Session test ở 17h50 làm bài B của codeforces contest #675 div 2 (Fair numbers).

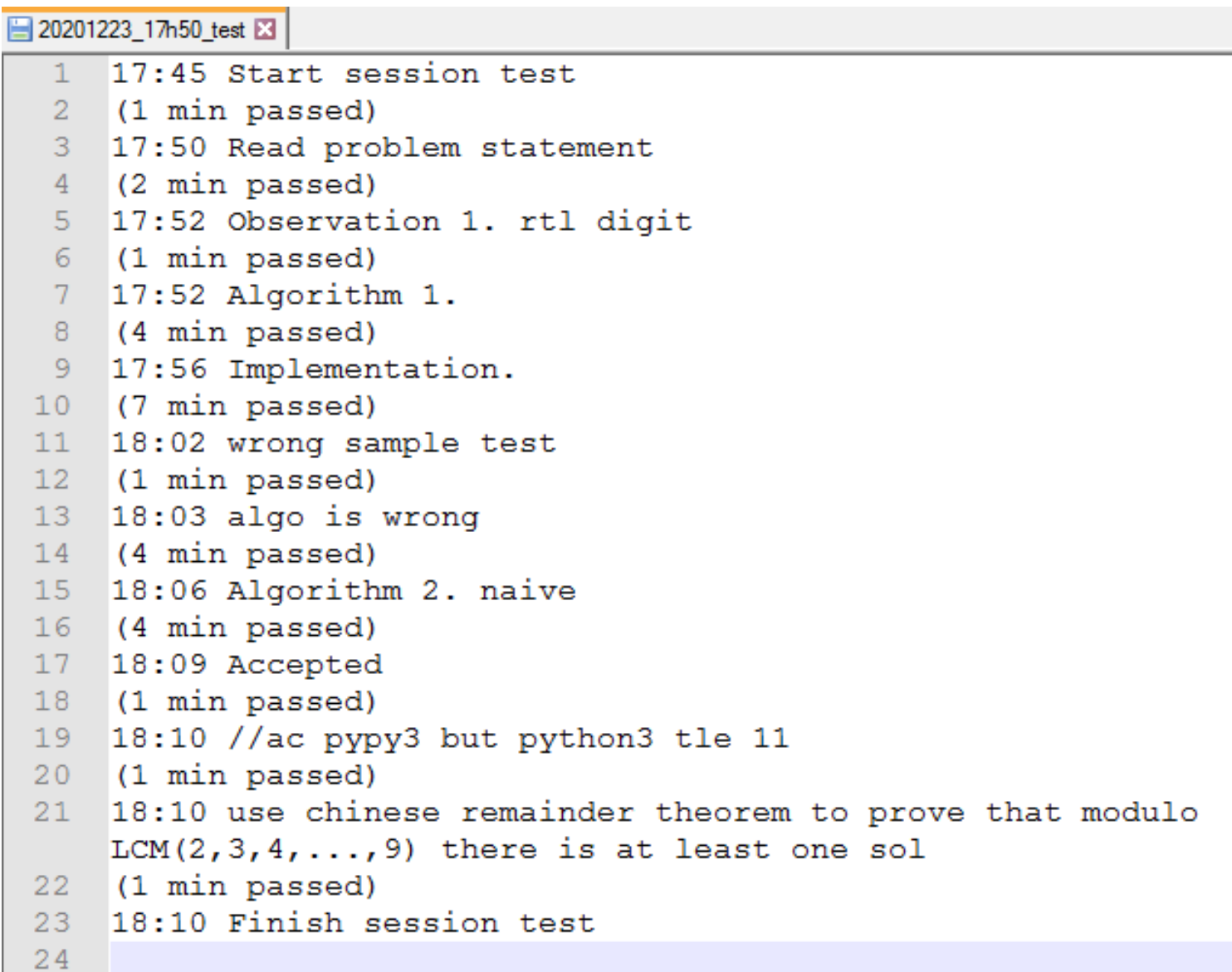
Bản đầu tiên nhiều đó thôi, được chưa?

```

na test
rd
ob rtl digit
al
im
wrong sample test
algo is wrong
al naive
ac
//ac pypy3 but python3 tle 11
use chinese remainder theorem to prove that modulo LCM(2,3,4,...,9) there is at least one sol
done
exit

```

Figure 1: Input 1



```

20201223_17h50_test x
1 17:45 Start session test
2 (1 min passed)
3 17:50 Read problem statement
4 (2 min passed)
5 17:52 Observation 1. rtl digit
6 (1 min passed)
7 17:52 Algorithm 1.
8 (4 min passed)
9 17:56 Implementation.
10 (7 min passed)
11 18:02 wrong sample test
12 (1 min passed)
13 18:03 algo is wrong
14 (4 min passed)
15 18:06 Algorithm 2. naive
16 (4 min passed)
17 18:09 Accepted
18 (1 min passed)
19 18:10 //ac pypy3 but python3 tle 11
20 (1 min passed)
21 18:10 use chinese remainder theorem to prove that modulo
LCM(2,3,4,...,9) there is at least one sol
22 (1 min passed)
23 18:10 Finish session test
24

```

Figure 2: Output 1 (nhìn tên file)

na test remaining features
go
al xtrace
db 2
db LCA not true
db dmm\nka
gg omae wa mou shindeiru
wa 7
tle 11
tle sample 4
mle 12
rte s4
ac
fuuuuuuuu
fu
tc
tc 2
tc s4
afk
bck
ed wheel factorization
ed
dm
Dit me may
not a special command

```
1 18:18 Start session test remaining features
2 (1 min passed)
3 18:18 Algorithm 1. xtrace
4 (0 min passed)
5 18:18 Debug 2
6 (1 min passed)
7 18:18 Debug LCA not true
8 (0 min passed)
9 18:18 Debug dmm\nka
10 (1 min passed)
11 18:18 Google omae wa mou shindeiru
12 (0 min passed)
13 18:18 Wrong answer test 7
14 (1 min passed)
15 18:18 Time limit exceeded test 11
16 (0 min passed)
17 18:18 Time limit exceeded sample 4
18 (0 min passed)
19 18:18 Memory limit exceeded test 12
20 (1 min passed)
21 18:18 Run time error s4
22 (0 min passed)
23 18:18 Accepted
24 (1 min passed)
25 18:18 fuck up
26 (1 min passed)
27 18:18 fuck up
28 (0 min passed)
29 18:18 Lookup test case
30 (1 min passed)
31 18:18 Lookup test case 2
32 (0 min passed)
33 18:18 Lookup test case s4
34 (1 min passed)
35 18:18 Away from keyboard.
36 (0 min passed)
```