

# Peter 笔记

Peterlits Zo

2020 年 6 月 4 日

## 目录

<b>1</b>	<b>6 月笔记</b>	<b>2</b>
1.1	To-do List . . . . .	2
<b>2</b>	<b>HDU-6400</b>	<b>2</b>
2.1	题目 . . . . .	2
2.2	想法 . . . . .	2
2.3	题解 . . . . .	3

## SECTION 1

## 6 月笔记

### 1.1. To-do List

☐ 重新做一下 HDU-6400

注: ☐ 未完成, ☒ 正在完成中, ☒ 已经完成。

## SECTION 2

Lua $\text{\LaTeX}$  的使用

今天为了解决计算时间太慢的问题,找了一下 Lua $\text{\LaTeX}$  的 tutorial,发现的确不错,对于我来说,只需要改动一点点地方就可以运行,能够完美支持`ctexart`的底层就是一个好底层。

而我最喜欢的就是:它本身可以运行 Lua 代码,而这个代码的速度是非常非常快的,作为一个脚本语言做到这么快还是不容易,而且标准库的算法很不错。(我就是因为,原来的宏包把时间转换为具体的时间很慢才找到 Lua $\text{\LaTeX}$  的,我翻译原来的那个,是每一次都从头加到尾,还要判断闰年什么的,不建表)。

其次,宏展开是优先于运行代码的,所以很容易很容易就可以向 Lua 传入参数,输出就用`luaact`(好像不是这么写的),输出到原文件中,就像用 JavaScript 来操纵 HTML 一样。太棒啦。

## SECTION 3

 $\text{\LaTeX}$  的长度

最近,在写  $\text{\LaTeX}$  宏包的时候,出现了一点幽灵长度,我总是搞不清楚,为什么高一点的和矮一点的间距不平等,后来我终于搞懂了,记录在案。

### 3.1. 段落、文字行之间的间距

一般来说,两个段落之间是由宏`\par`而隔开的,所以说那两个段落之间的竖直盒子是由`\par`命令决定的,而`\par`以变量,长度寄存器变量`\parskip`来决定,它的值是`0pt plus 1pt`,弹性长度

下限一定是固定的，但是上限不是固定了（当然如果超过上限太多了，那它就是一个坏盒子了）。

文字行之间的间距，这是由`\baselineskip`决定了。这就是说，如果我的这一行盒子都差不多是一个字符高的话，那么它乖乖地采用下限，留出好看的间隙，如果超过了，那它的高度就不是上限了，而是它本身的高度了。

## 3.2. 解决方案

定义`\baselineskip`为`0pt`，这样的话，无论是多高的话，它的高度就是它本身高度了，而不会留间隙，固定的间隙应该用`\vspace*`来代替它，用在两个命令之间，构建一个漂亮的间隙。

### SECTION 4

## HDU-6400 题解

杭州电大的 ACM 官网好慢哦，注册一下又要等半天。然后好像官网的题解也没有，有点难受。

## 4.1. 题目

这道题，说的是有一个括号矩阵，给定了  $n$  和  $m$  作为长和高，然后让行匹配和列匹配的个数最多。那么什么情况下是匹配的呢：一个正向括号就是匹配的，反之则不匹配。

比如：

1.	2.	3.
( ) ( )	(( ( ) ( ( ) ) ) )	(
		)
		(
		)

这三个分别有行匹配和列匹配的情况。计数为 1。我们要对于给定  $n$  和  $m$  而言选定计数最大的进行输出。很明显，这个不是唯一的。

## 4.2. 想法

在做这个之前我有一点想法，但是做错了。

首先对于正向括号序列而言，那么有：它的长度  $len$  一定满足：

$$len \pmod 2 = 0$$

所以说如果这个是一个奇数 (odd number) 的话, 无论如何也不可能是一个正向括号序列。所以有:

1.  $n$  和  $m$  都是奇数, 那么计数  $cnt$  一定是零。
2.  $n$  是奇数但是  $m$  不是, 那么计数  $cnt$  为  $n$  (因为  $n$  对应的那一个构造成正向括号的话, 计数  $cnt$  就为  $n$ , 这个时候是最优解)。反之亦然。
3. 我感觉最难的  $>_<$ , 那就是  $n$  和  $m$  都是偶数。这种情况下我想了一个绝妙的想法那就是, 令  $a = \max(n, m)$ ,  $b = \min(n, m)$ , 那么有如果我把  $a$  对应的行/列都改成正向括号序列, 那么答案计数  $cnt$  就是  $a$  了。但是不够, 后来我有想到:

```
(()) => (())
()() => () ()
(()) => (())
()() => () ()
```

像这种的话, 就太棒了! 首先它本身有四个正向序列构成。还见缝插针的搞了一个竖的 (第二列) 所以说, 只要我在原有基础上多插插就有多的了。这个时候, 计数  $cnt$  的值就为  $a + \frac{b}{2} - 1$ , 其中  $\frac{b}{2}$  是因为原第一排只有一半是开括号, 可以用来构建正向序列, 减一是因为第一个开括号是不能搞事情的。

### 4.3. 题解

只能说我想到了一点但是没有想到第二点。这道题需要构建正向括号序列, 但是其实正向的它的本质是开头是 ( 末尾是 ) 然后其他的只要保证中间不跌下到 -1 (正向序列的值, 意思是有没有匹配上的) 反括号) 并且末尾为零就好了。

那么这么看的话,

```
.(((((. => .(((((.
(. . . . .) => ()()()()
(. . . . .) => (())()()
(. . . . .) => ()()()()
(. . . . .) => (())()()
.))))) . => .))))) .
```

上面的才是最优解, 只要它中间那一坨能够满足的话整个的计数就是  $cnt = a + b - 4$ , 两个式子比较有

$$(a + b - 4) - (a + \frac{b}{2} - 1) \quad (1)$$

$$= b - \frac{b}{2} - 3 \quad (2)$$

$$= \frac{b}{2} - 3 \quad (3)$$

$$> 0 \quad (4)$$

$$\implies b > 6 \quad (5)$$

, 所以说在大一点的时候应该用这个方法会比较好一点。