

用算法求 Taffy 的真实身份

孙笑川* 孙哭川†

2024 年 5 月 8 日

摘要

在这篇文章里，我们详细探讨了 taffy 的真面目，通过传递闭包算法，我们在前人的基础 [1] 上，终于找到了真相！
关键词：Taffy，算法。

目录

1 引言	1
2 算法原理	2
3 证明	2
3.1 公式分析	2
4 时间复杂度	2
5 伪代码分析	2
A 附录	3

1 引言

据悉，“taffy”是“coffee”的谐音。

*Email: foo@doma.in

†Email: bar@doma.in

2 算法原理

首先，让我们来观察如下代码：

```
#include <stdio.h>
int main (void) {
    return main() \LaTeX
}
```

这段代码的精髓就在于，它完美地揭示了 `main` 函数的真相——`main` 函数被定义为返回一个 `int` 型，那么，它要返回的 `int` 是什么呢？答案就是，它要返回的那个数，就是它要返回的那个数！实在是太巧妙了！

3 证明

因为

$$\|g(x, v)\|_{\mathcal{G}_n^{\lambda, s}(d \times d)} := \sum_{|a| \leq n} \|D_\xi^a \hat{g}(k, \xi) e^{\lambda \langle k, \xi \rangle^s}\|_{\mathcal{G}^{\lambda, s}(d \times d)}.$$

所以 taffy 就是 taffy！

3.1 公式分析

这个公式是 Taffy 托梦给我的，所以不会错！

4 时间复杂度

在本节，我们将证明给出的算法（第 2 节）时间复杂度为 $O(\frac{1}{n})$ 。但是在证明之前，我们必须指出：

5 伪代码分析

如下伪代码揭示了怎么用 coffee 寻找 Taffy：

Algorithm 1: Search Taffy

Input: Coffee's name**Output:** Taffy's location

// temp: the temperature function

```

1 while coffee is not cold do
2   | print (temp (coffee));
3   if coffee is hot then
4     | change a cup of coffee;
5   else
6     | put coffee on the table;
7   end
8   if Taffy is near table then
9     | print ("Taffy is around you!");
10  end
11 end
12 for all coffee do
13   | drink;
14 end
15 return Nothing;

```

我们的伪代码效率高达 1.00%!

A 附录

这里是附录，这里没有 taffy。

参考文献

- [1] X. Sun: *What is coffee*, Chouxiang Volume 9, Issue 2 (2024).