

The Best Title Ever

Sheldon Cooper Hays Whitlatch

December 4, 2019

Abstract

We should really write an abstract. We should really write an abstract.
We should really write an abstract. We should really write an abstract.
We should really write an abstract. We should really write an abstract.
We should really write an abstract. We should really write an abstract.
We should really write an abstract. We should really write an abstract.
We should really write an abstract. We should really write an abstract.
We should really write an abstract. We should really write an abstract.
We should really write an abstract. We should really write an abstract.
We should really write an abstract.

1 Introduction

Definition 1. This is a definition of laziness. This is a definition of laziness.
This is a definition of laziness. This is a definition of laziness. This is a definition
of laziness. This is a definition of laziness. This is a definition of laziness. This
is a definition of laziness. This is a definition of laziness.

Definition 2. Here we define procrastination. Here we define procrastination.
Here we define procrastination. Here we define procrastination. Here we define
procrastination. Here we define procrastination. Here we define procrastination.
Here we define procrastination. Here we define procrastination. Here we define
procrastination.

Now we can reference Definition 1 and Definition 2. Notice that this is the
beginning of a new paragraph so it is indented. You can always get rid of the
indentation by using noindent (see next paragraph).

Now we can reference Definition 1 and Definition 2. Notice that this is the
beginning of a new paragraph so it is indented. You can always get rid of the
indentation by using noindent (see this paragraph).

This is how we cite: The best solutions appear in [Bab16] and in [BW91]. I
have a separate file called bibliography.bib. In there I put all the papers that I
will (probably) cite. There are several ways to get the citation format (including
some from MAA.org), however since I am lazy (see Definition 1), I use google
scholar.

Lemma 1 ([Bab16], **Theorem 9**). *This is a theorem that appears in [Bab16]. Here it appears as Lemma 1 but in [Bab16] it is Theorem 9. I included this info so that the reader can easily find a proof of it (which I omit here)*

I'm hungry, time for an apple:



Figure 1: A pretty picture

Example 1. *Here is an example so that you can better understand. Here is an example so that you can better understand. Here is an example so that you can better understand. Here is an example so that you can better understand. Here is an example so that you can better understand. Here is an example so that you can better understand.*

If you ignore the laws of algebra then you can prove just about anything.

Theorem 1. *Let $\epsilon > 0$ be given. There exists $\delta > 0$ and $n \in \mathbb{N}$ so that*

$$2 = 1$$

Proof. Let $x = y$

$$\begin{aligned}x &= y \\(x - y)(x + y) &= x(x - y) \\x + y &= x \\2x &= x \\2 &= 1\end{aligned}$$

□

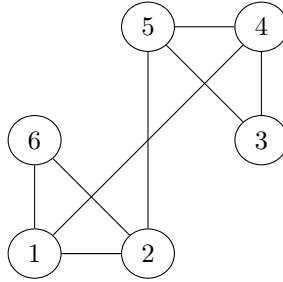
Observation 1.

The above proof can be extended to show that $k = 1$ for any $k \in \mathbb{N}$.

Let's start on a new page.

2 Section Meow

I made a tikz drawing



I would like some space before and after this.

An example of pseudo-code in latex:

Algorithm 1

```
1: input: a directed acyclic graph  $D$ .
2: output: true or false. True if the transitive closure of  $D$  is a  $\mathcal{V}$ -poset,
   False otherwise.
3: Bool  $\leftarrow$  true
4: if IsSeriesParallel( $D$ )[Bool]=False then
5:   Bool  $\leftarrow$  false
6: else
7:   D  $\leftarrow$  IsSeriesParallel( $D$ )[DAG]
8:   if IsBowtieFree( $D$ )= false then
9:     Bool  $\leftarrow$  false
10:  else
11:    if ClosureIsVPoset( $D$ ) = false then
12:      Bool  $\leftarrow$  false
13:  return Bool
```

3 Future Work

Question 1. *Here is an open question that reader should be interested in...*

Conjecture 1. *Here is an open question that reader should cite me in their solution...*

References

- [Bab16] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 684–697. ACM, 2016.
- [BW91] Graham Brightwell and Peter Winkler. Counting linear extensions. *Order*, 8(3):225–242, 1991.