

Contents

1	Introduction	2
2	Setup Environment	2
2.1	Operating System	2
2.2	Install texlive (a TeX distribution) **	2
2.2.1	Terminal Command	2
2.2.2	More Detailed Instruction	2
2.3	Install g++ compiler **	2
2.4	Download Source Code **	2
3	How to Use	2
4	Program Behaviour	3

1 Introduction

Updating code library before every onsite contest is a nightmare. A brute force solution is presented here. This procedure takes all source files from given location and makes a tex file, then compiles that file using a tex distribution and generates a cool looking pdf file.

2 Setup Environment

2.1 Operating System

This program only works on linux distribution because it uses some system calls which are not available on windows.

2.2 Install texlive (a TeX distribution) **

[Instead, you can use this online tool <https://www.sharelatex.com>]

2.2.1 Terminal Command

```
sudo apt-get install texlive-full
```

2.2.2 More Detailed Instruction

<https://milq.github.io/install-latex-ubuntu-debian/>

2.3 Install g++ compiler **

Terminal Command

```
sudo apt-get install g++
```

2.4 Download Source Code **

<https://ideone.com/X2me9X>

3 How to Use

Run the source code (Codebocks/Terminal both works fine). This program scans two input string from standard input stream. First one is absolute location, that contains all your souce codes. Second one is the file name, that you want to save as. If no error occures, outputs (.tex, .pdf, .log, .aux, .toc) will be saved in 'tex_output' under 'Documents' directory.

4 Program Behaviour

- Notice that your given source codes' location can contain both sub-folders and source code files. Folder name will be used as section name like given screenshot.

Contents	
Algorithm	1
1 Algorithm Implementations	2
1.1 Backtracking	2
1.1.1 N Queens Chess Problem	2
1.2 Big Integer Library	5
1.2.1 addition	5
1.3 Data Structure	7
1.3.1 Fenwick Tree (2D)	7
1.3.2 Fenwick Tree	9
1.3.3 Graph Connectivity (Disjoint Set)	11
1.3.4 Lowest Common Ancestor	14
1.3.5 Network Connections (Disjoint Set)	16
1.3.6 Palindromic Tree	19
1.3.7 Postfix Conversion and Evaluation	20
1.3.8 Segment Tree (2D)	23
1.3.9 Segment Tree (Lazy Propagation)	27
1.3.10 Segment Tree (Lazy Propagation, push down)	28
1.3.11 Segment Tree	31
1.3.12 Square Root Decomposition	34
1.3.13 Trie	37
1.3.14 Virtual Friends (Disjoint Set)	39
1.4 Dynamic Programming	42
1.4.1 Balanced Partition Possibility	42
1.4.2 Knuth Optimization	44
1.4.3 LCS	46
1.4.4 Minimum Vertex Cover	49

Image: Generated Pdf

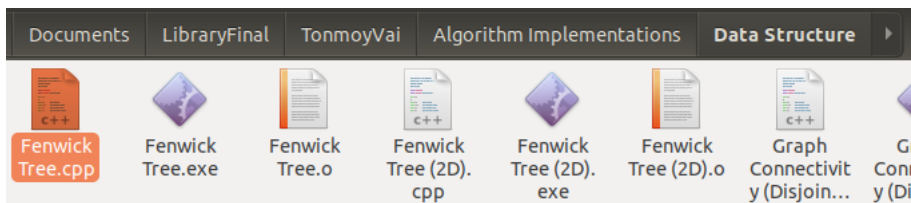


Image: Path looks Like

- If a source code is located at 4 layer (by layer I mean folder under folder) further than given location, that file will be ignored.
- Files that do not have any of .cpp, .py, .c, .java extensions will be ignored.
- Highly Recommended not to use special characters in source file name and anywhere of code.
- Ignore White Spaces in path name and file names.
- Program may contain bugs.