

Contents

1	File tree	1
2	SRC	4
2.1	Main	4
2.2	Process	4
2.3	Process file	4
2.4	Misra Gries	4
2.5	Sampling	4
2.6	Vote generator	4
2.7	Vote rules	4
2.7.1	Graph + graph voting	4
2.7.2	Vector + plurality	4
2.7.3	STV	4
2.8	Error finders	4
2.8.1	Copeland error	4
2.8.2	Maximin error	4
2.8.3	Plurality error	4
2.8.4	STV error	4
2.9	Preflb vote parsers	4
3	Test	5

1. File tree

```
.
+-- docs
|   +-- developer_documentation
|   |   +-- developer_documentation.pdf
|   |   +-- developer_documentation.tex
|   +-- specification
|   |   +-- specification.pdf
|   |   +-- specification.tex
|   +-- user_documentation
|   |   +-- error_finder.png
|   |   +-- generate_votes.png
|   |   +-- run_algorithm.png
|   |   +-- save_as.png
|   |   +-- user_interface_main.png
|   |   +-- user_documentation.pdf
|   |   +-- user_documentation.tex
+-- README.md
+-- src
|   +-- error_finders
|   |   +-- copeland_error.py
|   |   +-- maximin_error.py
|   |   +-- plurality_error.py
|   |   +-- stv_error.py
|   +-- main.py
|   +-- misra_gries.py
|   +-- preflib_vote_parsers
|   |   +-- copeland_parse.py
|   |   +-- maximin_parse.py
|   |   +-- plurality_parse.py
|   |   +-- stv_parse.py
|   +-- process_file.py
|   +-- process.py
|   +-- sampling.py
|   +-- vote_generator.py
|   +-- vote_rules
|   |   +-- brute_force
|   |   |   +-- graph.py
|   |   |   +-- graph_voting.py
|   |   |   +-- plurality.py
|   |   |   +-- stv.py
|   |   |   +-- vector.py
+-- test
    +-- all_test.py
    +-- graph_voting_test.py
    +-- misra_gries_test.py
    +-- plurality_test.py
    +-- sampling_test.py
    +-- stv_test.py
```

This is whole file tree of the project, but we will focus just on "src" and "test" folders. That means this:

```
+-- src
|   +-- error_finders
|   |   +-- copeland_error.py
|   |   +-- maximin_error.py
|   |   +-- plurality_error.py
|   |   +-- stv_error.py
|   +-- main.py
|   +-- misra_gries.py
|   +-- preflib_vote_parsers
|   |   +-- copeland_parse.py
|   |   +-- maximin_parse.py
|   |   +-- plurality_parse.py
|   |   +-- stv_parse.py
|   +-- process_file.py
|   +-- process.py
|   +-- sampling.py
|   +-- vote_generator.py
|   +-- vote_rules
|       +-- brute_force
|       +-- graph.py
|       +-- graph_voting.py
|       +-- plurality.py
|       +-- stv.py
|       +-- vector.py
+-- test
    +-- all_test.py
    +-- graph_voting_test.py
    +-- misra_gries_test.py
    +-- plurality_test.py
    +-- sampling_test.py
    +-- stv_test.py
```

2. SRC

2.1 Main

Used for handling the input interface so most of the functions are for generating windows with Tkinter.

2.2 Process

This class is basically connector between all the other classes, it calls all the algorithms and connect them with the interface.

2.3 Process file

2.4 Misra Gries

2.5 Sampling

2.6 Vote generator

2.7 Vote rules

2.7.1 Graph + graph voting

2.7.2 Vector + plurality

2.7.3 STV

2.8 Error finders

2.8.1 Copeland error

2.8.2 Maximin error

2.8.3 Plurality error

2.8.4 STV error

2.9 Preflb vote parsers

All the parsers work basically the same they initialize the vote rule and convert the lines of Preflb file into something that the vote rules implementation understand.

3. Test

Tests are basic unit tests, that looks something like this:

```
1 import unittest
2 from src.vote_rules.brute_force.stv import stv
3
4 def test_stv1(self):
5     candidates = [1, 2, 3, 4, 5]
6     votes = [[4, 2, 3, 1, 5], [3, 1, 2, 4, 5], [4, 1, 3, 5, 2], [5, 1, 3, 2,
7         4]]
8     S = stv(candidates, votes)
9     expected_output_1 = [4, 3, 5, 1, 2] # Expected output dictionary
10    self.assertEqual(S.stv(), expected_output_1)
```

The only not so easy tests were tests for sampling, since sampling is randomized, so we needed to add something called error margin and run it on big enough data,. That basically means, how much error we allow, so we let error margin be 0.09.