Listings

| 1 | 这是个生成器 | 2 |
|---|---------------------------|---|
| 2 | cpp-code/huffman copy.cpp | • |
| 3 | Huffman Tree | : |

Listing 1: 这是个生成器

```
1
   from pathlib import Path
2
   import json
3
   import argparse
4
   parser = argparse. ArgumentParser(description="Convert_code_in_current_directory
       \sqcup to \sqcup a \sqcup latex \sqcup document.")
6
   parser.add_argument('output', help="output_path")
7
8
   args = parser.parse_args()
9
10
   output_path = args.output
11
   out_file = open(output_path, 'w', encoding='utf-8')
12
13
   template\_str = """
14
   15
16
17
   template_param_str = ",\{0\}=\{1\}"
18
19
   begin_code = """
20
21
    | | document class \{ report \} |
22
    | | usepackage \{ geometry \} |
23
    | geometry \{a4paper, scale = 0.8\}
24
   | usepackage \{xeCJK\}
25
   | | usepackage \{ listings \} |
26
   | | lstset |
27
        numbers=left,
28
        frame = single,
29
        caption = | | lstname |
30
        breaklines=true
31
32
   | | begin \{document\} |
33
   \setminus \setminus lstlistoflistings
34
35
   | | newpage
36
    ,, ,, ,,
37
   end code = """
38
    39
    " " "
40
41
42
   ext_filters = { '.c': 'c', '.cpp': 'c++', '.h': 'c++', '.hpp': 'c++', '.java': '
       java', '.py': 'python'}
43
   out_file.write(begin_code)
44
45
46
   p = Path(', ')
   filenames = list(p.rglob('*'))
47
   for f in filenames:
48
        fs = str(f)
49
50
        ext = fs [fs.rfind('.'):]
        if (ext not in ext_filters):
51
52
            continue
        param_str = ""
53
```

```
54
       if (Path(fs + '.json').is_file()):
55
            with open(fs + '.json', 'r', encoding='utf-8') as config_file:
56
                file_content = config_file.read()
                params = json.loads(file_content)
57
                for k in params:
58
                    param\_str += template\_param\_str.format(k, params[k])
59
60
       fs = fs.replace('\\', '/')
61
       out_file.write(template_str.format(ext_filters[ext], fs, param_str))
62
63
   out file.write(end code)
64
65
   out_file.close()
```

Listing 2: cpp-code/huffman copy.cpp

```
#include <iostream>
1
2
   // just for test
3
4
5
   struct Huffman
6
7
        int a = 0, b = 0, c = 0;
8
9
   };
10
11
   int main()
12
   {
        std::cout << Huffman().a;
13
14
        return 0;
15
```

Listing 3: Huffman Tree

```
#include <iostream>
1
2
   // just for test
3
4
5
   struct Huffman
6
7
        int a = 0, b = 0, c = 0;
8
9
   };
10
11
   int main()
12
   {
13
        std::cout << Huffman().a;
        return 0;
14
15
```