

# The Manual of the Mini Search Engine

---

First of all, welcome to evaluate our project.

Before starting the evaluation, I hope you can read the following content, which may solve some of your problems:

1. Please make sure that you are using a **Unix-like system**, such as Linux or MacOS, otherwise our program may not run properly (for example, it will be interrupted by an error message that a certain file cannot be opened)
2. Enter the code folder of this project in the CLI (command line interface) and enter the "make" command to compile all programs written in C/C++ and it will generate 4 executable files in the code directory, they are:

- **miniSearchEngine**: search engine program that provides search function
- **searchTest**: a test program for testing search capabilities. We have provided you with 6 test samples in advance, located in the code/source/search\_test directory, and you can freely modify the contents of the files. When running the program, enter the serial number after input to load it.

If you need to add new test sample files, please also put the files in this directory, and start with "input" and end with a serial number.

- **invIndexTest**: A program for testing the correctness of the inverted index. We provide command line parameters that can specify different modes for testing. For details, see part 2
  - **getStopWord**: A program for getting stop words, which will generate four files that our project depends on.
3. We use the python program to crawl the txt source file, which is located in the code/scripts/pycode folder. You can run the program. But please note that we assume that you are running the program in the code directory. If you run it in other directories, please modify the path in the program.

## Test on Inverted Index

---

To test the inverted index, you should follow the steps below:

1. Open the CLI(Command Line Interface, such as cmd、 powershell or other terminals), change the directory(cd) to `code` .
2. Compile the inverted-index-test file: input the instruction

```
make
```

then you will get the executable file called `invIndexTest` .

3. Run the program: input the instruction

```
./invIndexTest
```

If everything is OK, you will see the info below:

```
./invIndexTest
Now building an inverted Index:
Please input the directory of the documents:
Path:
```

You should input the correct directory containing some documentation(It's recommended that the directory is positioned in the `code` directory(the current directory)), like `tests/input3` (no quotation mark!).

Result:

```
./invIndexTest
Now building an inverted Index:
Please input the directory of the documents:
Path: tests/input3
Build successfully!
```

Failure(the incorrect directory):

```
./invIndexTest
Now building an inverted Index:
Please input the directory of the documents:
Path: haha
Could not open directory: No such file or directory
Fail to build an inverted index!
```

4. We have offered some useful CLI parameters for you to check more information about our inverted index. Here are the specific function of these parameters:

- `-f=n` or `--find=n` : You can search **single** word per time, and you totally have n times. If you omit `=n` , like `-f` , you have 1 time(default value)

```
./invIndexTest -f=3
Now building an inverted Index:
Please input the directory of the documents:
Path: tests/input3
Build successfully!

Finding Words Mode(only supports single word finding):
Find 1: great
Successfully find the word!
The word was found in files below:
1henryiv.1.3.txt: 6 times
Frequency: 6
-----
Find 2: haha
Sorry, no such word in the inverted index!
-----
Find 3: you
Sorry, no such word in the inverted index!
-----
```

- `-p` or `--print` : Print the whole B+ trees containing the inverted index. **You should use this parameter only when the word count is small.**
- `-s` or `--stopwords` : If you want to involve all stopwords to the inverted index, you can add this parameter to your command.
- `-t` or `--test` : Instead of make the traversal in the directory, you can make an inverted index just for a single document in the particular directory(the default directory is `tests` )

```
./invIndexTest -p -t
Now testing the correctness of inverted Index:
Please input the name of the input sample file:
Name: input2.txt
Build successfully!
B+ Tree of Inverted Index:
[ox, pig]
[ant, bear, cat, dog][ox, pear][pig, snake, whale]
```

- o `-tr` or `--time` : Record the duration for both the building of the inverted index and search a single word.

```
./invIndexTest -f -tr
Now building an inverted Index:
Please input the directory of the documents:
Path: docs
Build successfully!
Ticks: 2152143
Duration: 2.15s

Finding Words Mode(only supports single word finding):
Find 1: ice
Successfully find the word!
The word was found in files below:
allswell.2.3.txt: 1 time
asyoulikeit.3.4.txt: 1 time
# Delebrate Omission
troilus_cressida.3.3.txt: 1 time
two_gentlemen.3.2.txt: 1 time
Frequency: 18
-----
Ticks: 102
Duration: 0.00s
```