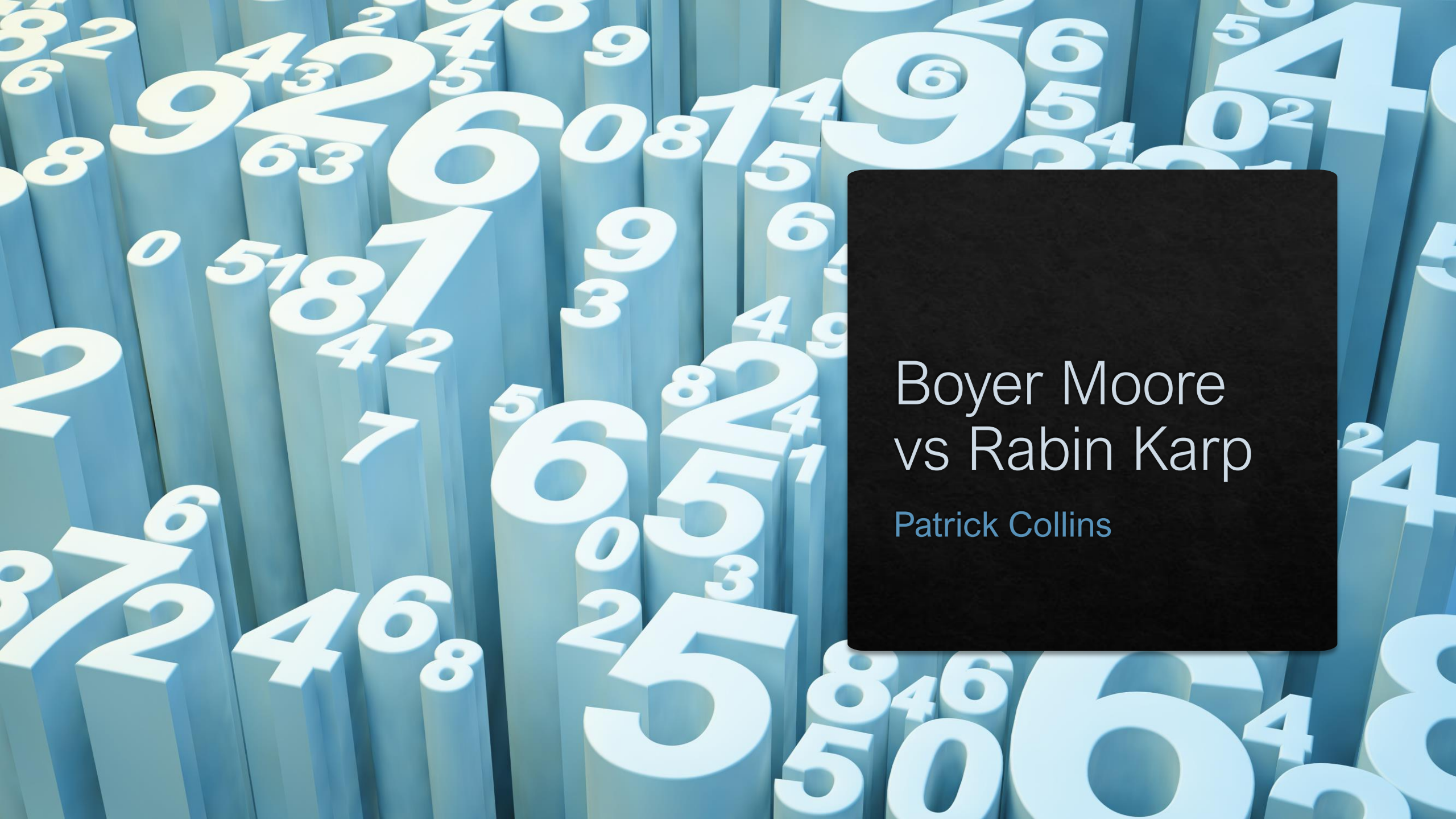


Boyer Moore vs Rabin Karp

Patrick Collins

Problem that I'll be solving

- Searching desired text within a text file, and returning matches and time.
- I will do this by implementing the Boyer-Moore-Horspool and Rabin Karp(without rolling hash) string searching algorithms.
- Once both algorithms can find desired text, I will compare them to find out which is better for this problem.

Data Structures

- For storing matches in both algorithms I will be using a list. Inserting at the back using push_back() as this will be constant time O(1).

- For Boyer Moore, the skipping will be done using an array as accessing the contents will be constant time O(1).

Expected Results

Goals

Boyer Moore should be much faster as text/input gets bigger

Rabin Karp should be affected by length of pattern.

Box plots

Boyer Moore's boxplots should be relatively close together, as not much variation between input length.

Rabin Karp's boxplots should vary as input size gets bigger.

Time complexity

Boyer-Moore-Horspool: Best case O(N/M), worst-case of O(NM).

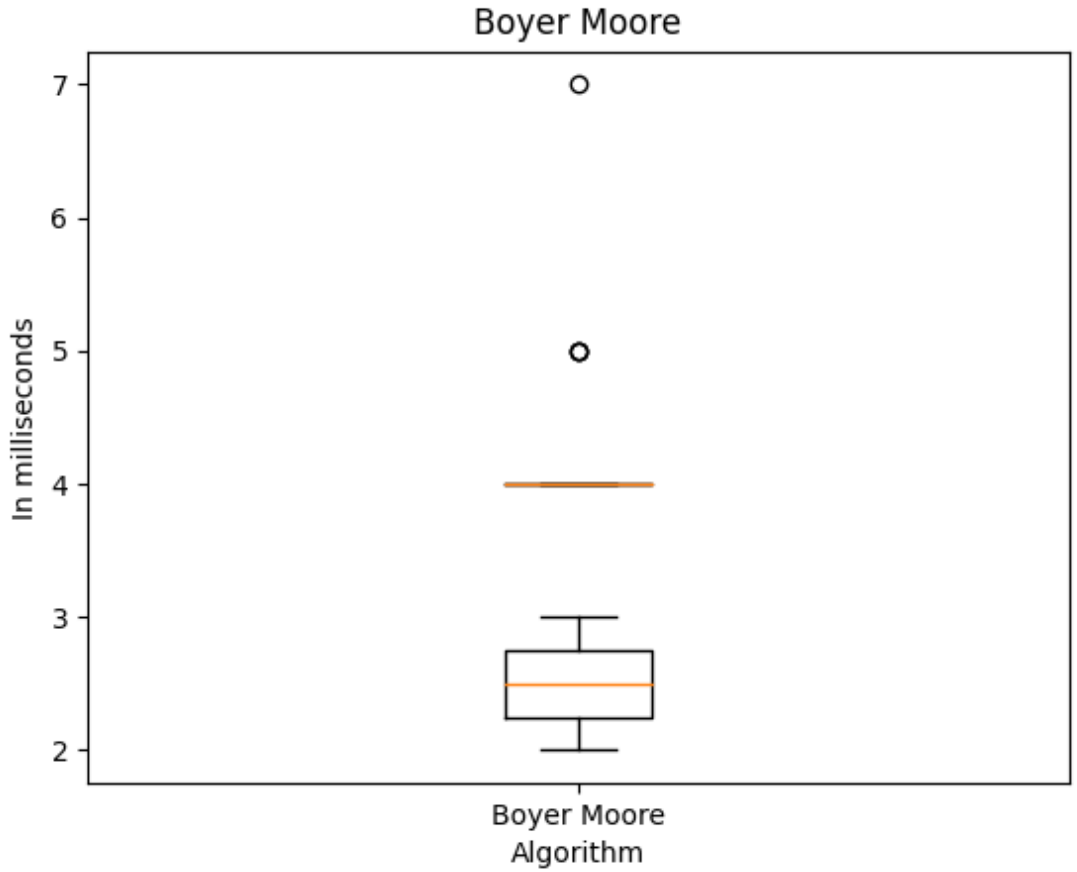
Rabin Karp: Best typical case O(N), worst case is also O(N)

Smaller text file

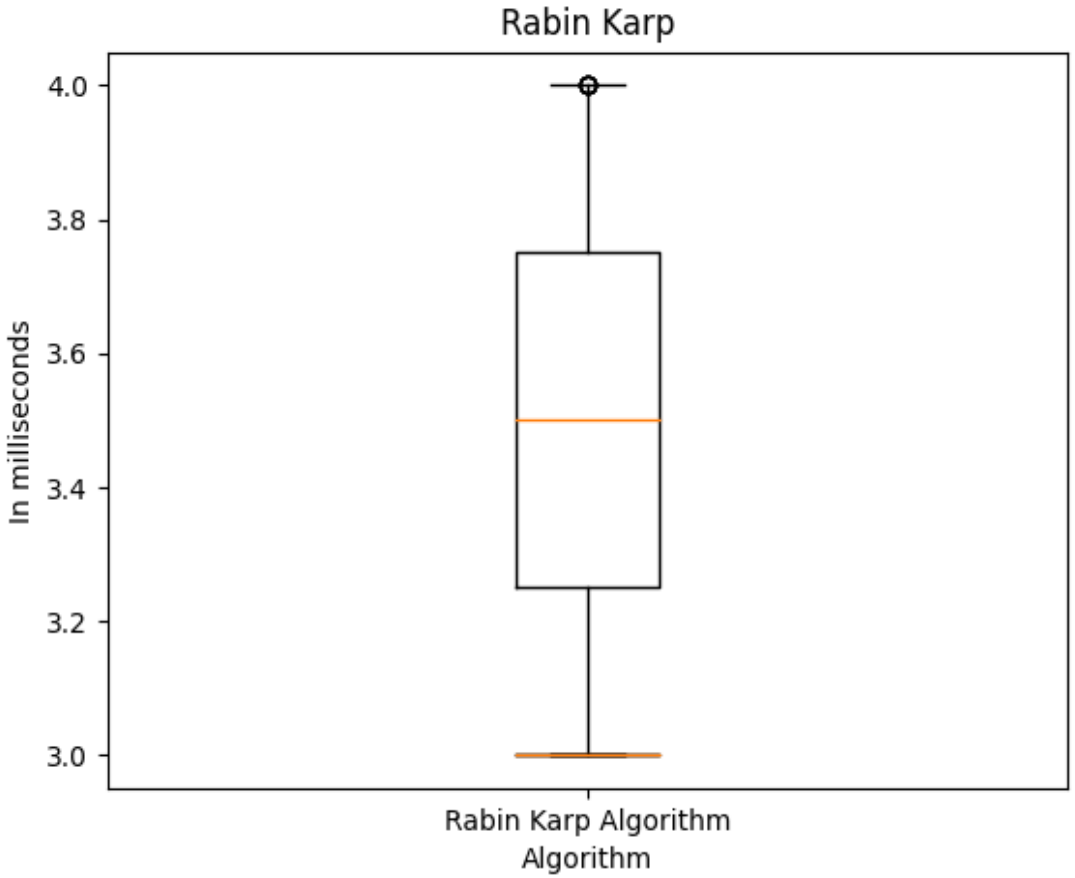
```
void load_jute_book(string& str) {
    // Read the whole file into str.
    load_file("jute-book.txt", str);

    // Extract only the main text of the book, removing the Project Gutenberg
    // header/footer and indices.
    str = str.substr(0x4d7, 0x2550c - 0x4d7);
}
```

Pattern=the (1000 iterations)

Boyer-Moore-Horspool

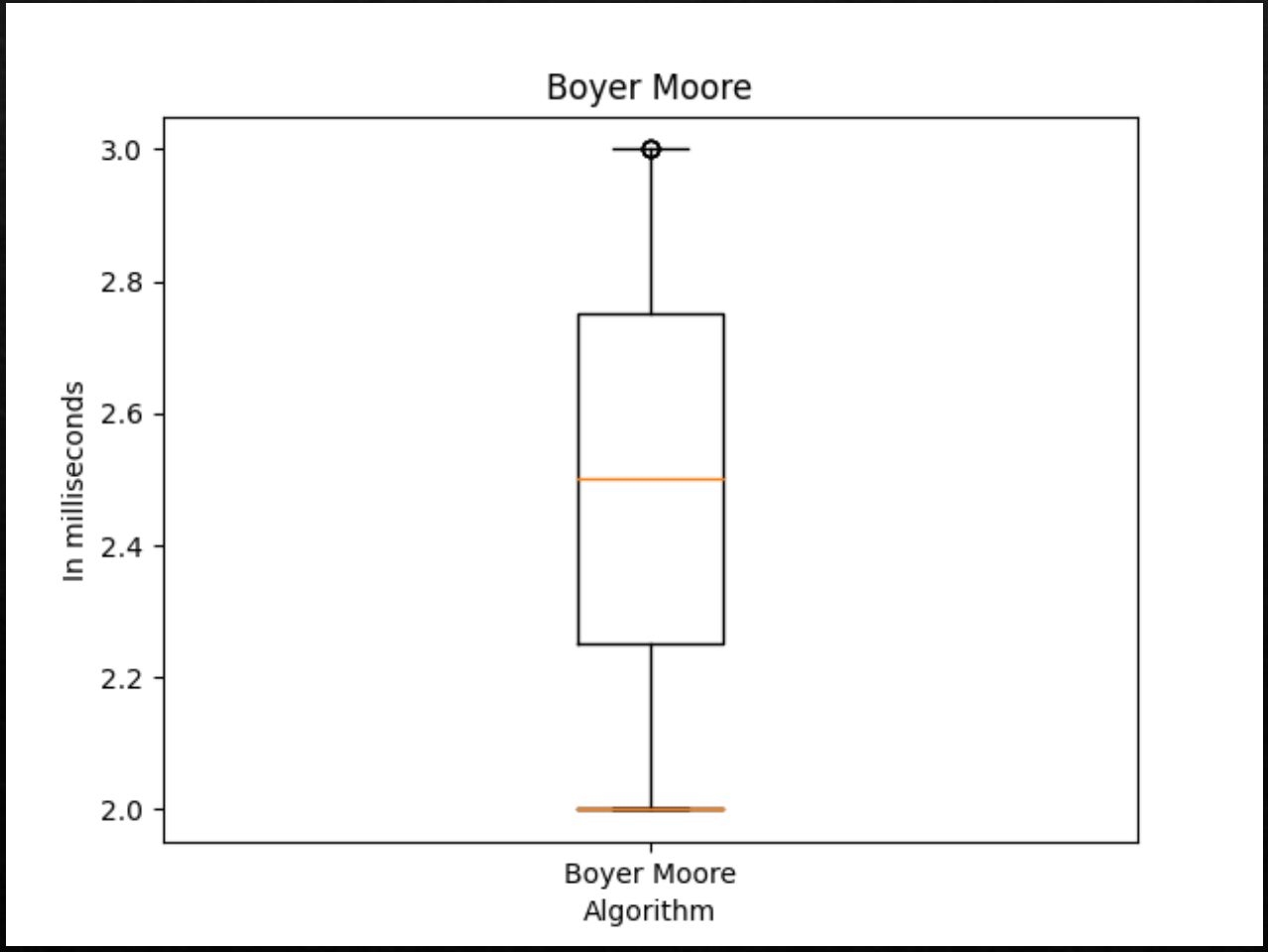


Rabin Karp

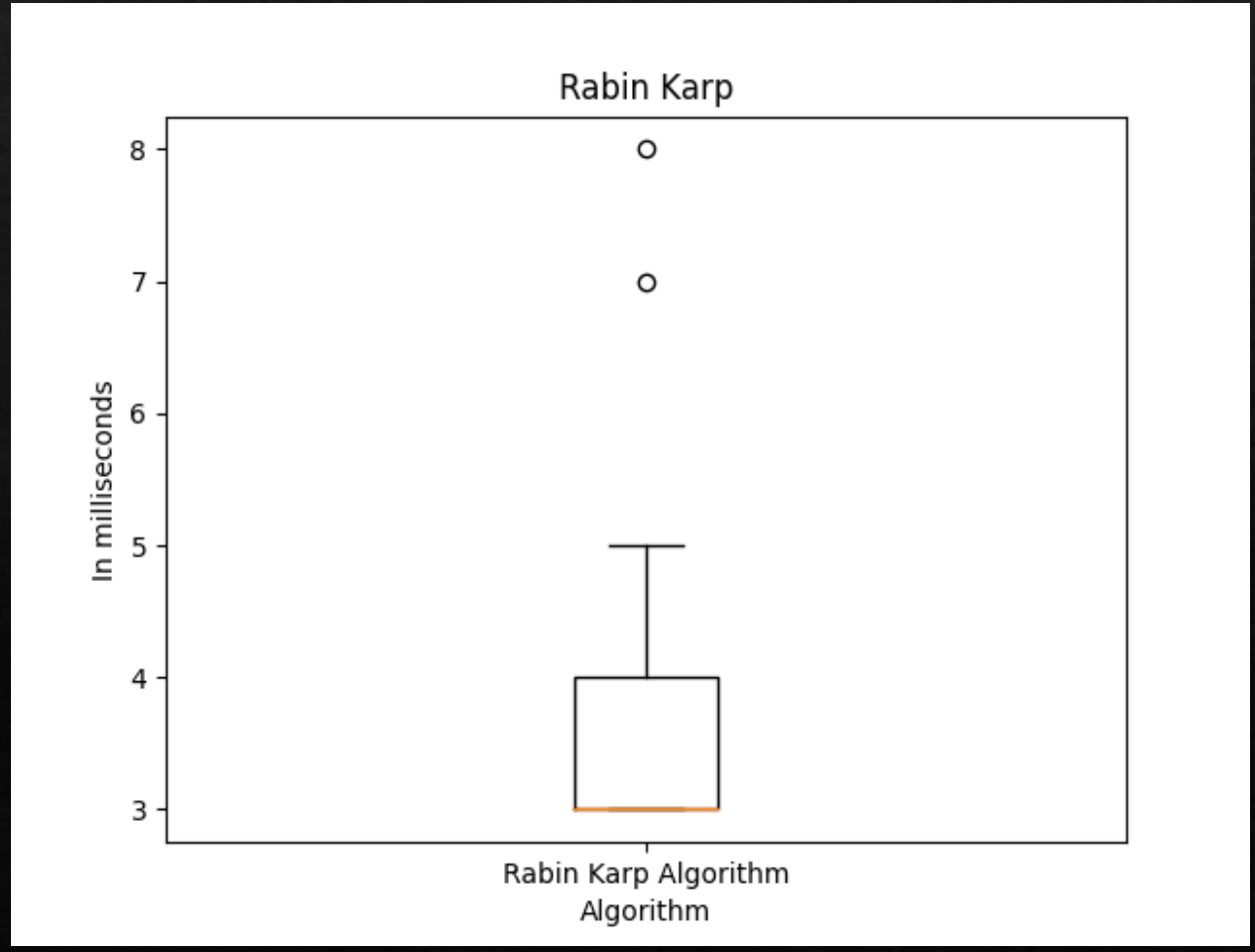


Pattern=Dundee (1000 iterations)

Boyer-Moore-Horspool

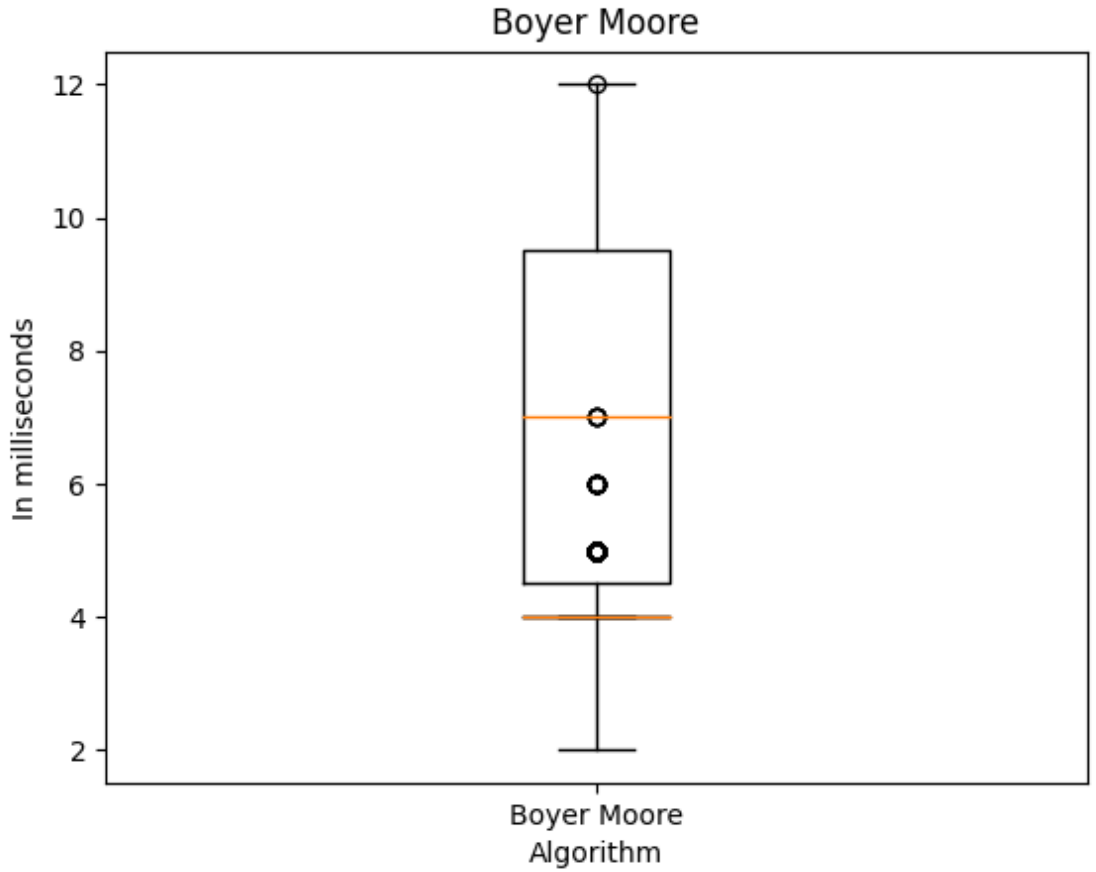


Rabin Karp

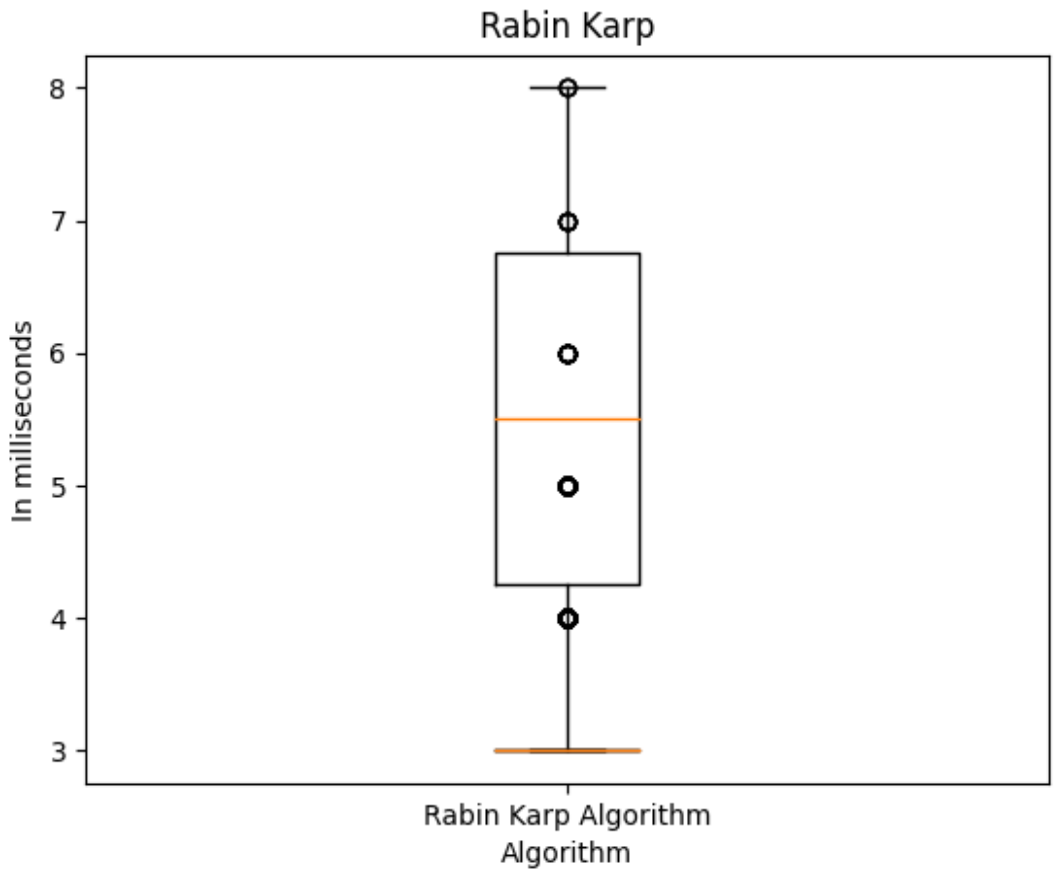


Pattern=the (10000 iterations)

Boyer-Moore-Horspool

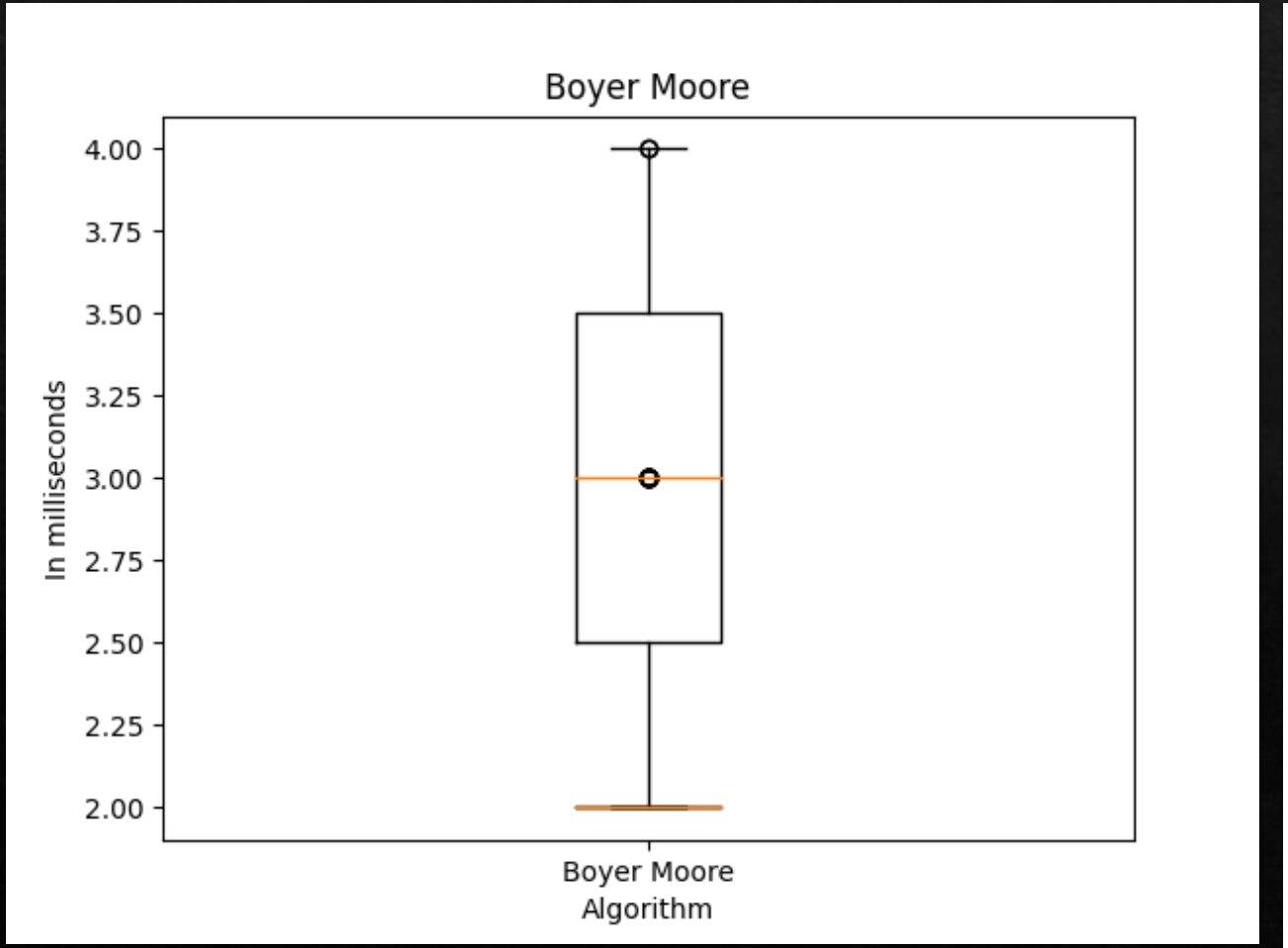


Rabin Karp

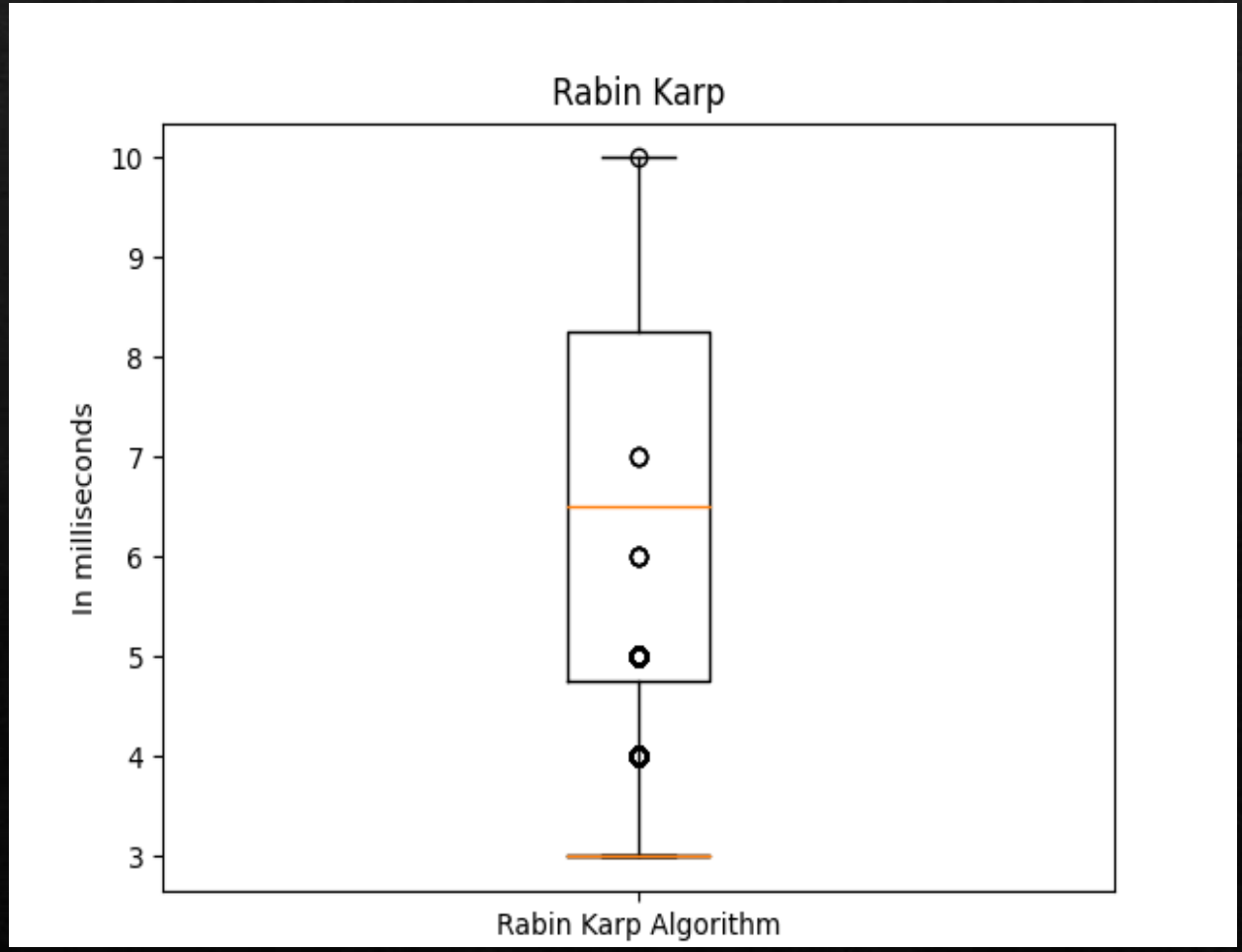


Pattern=Dundee (10000 iterations)

Boyer-Moore-Horspool



Rabin Karp

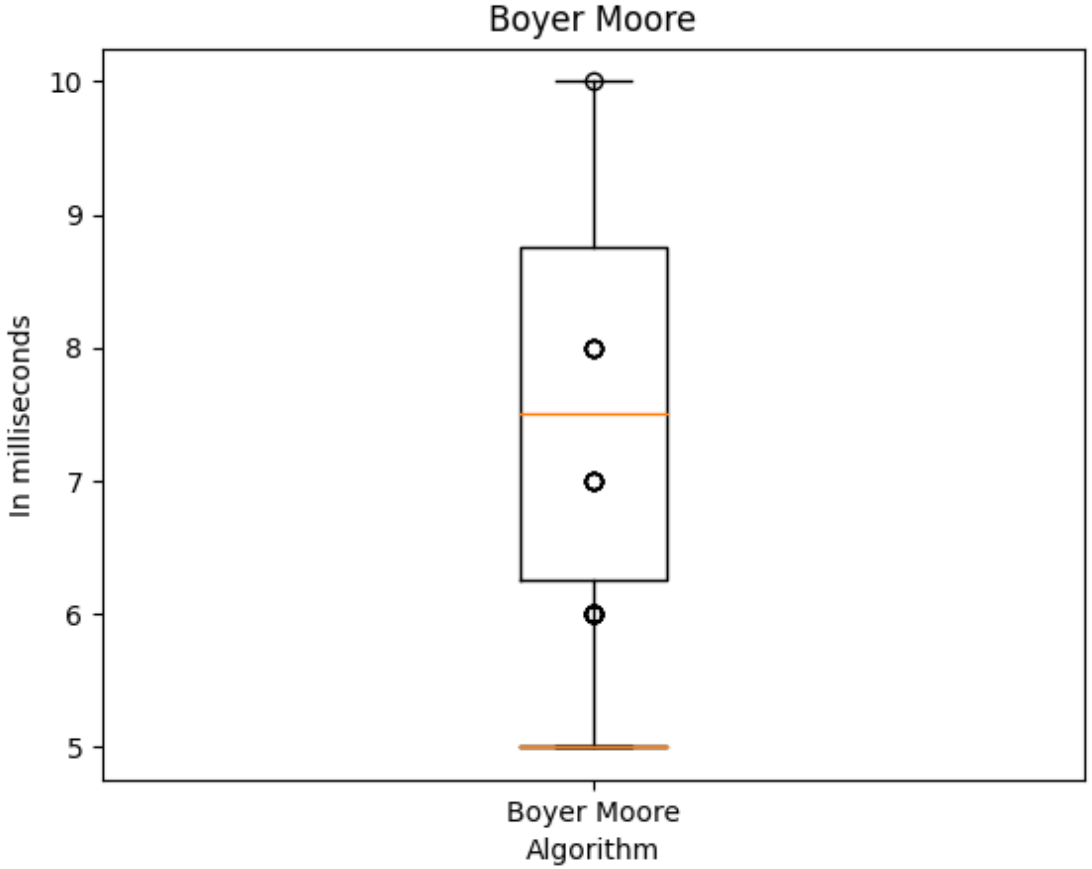


Bigger text file

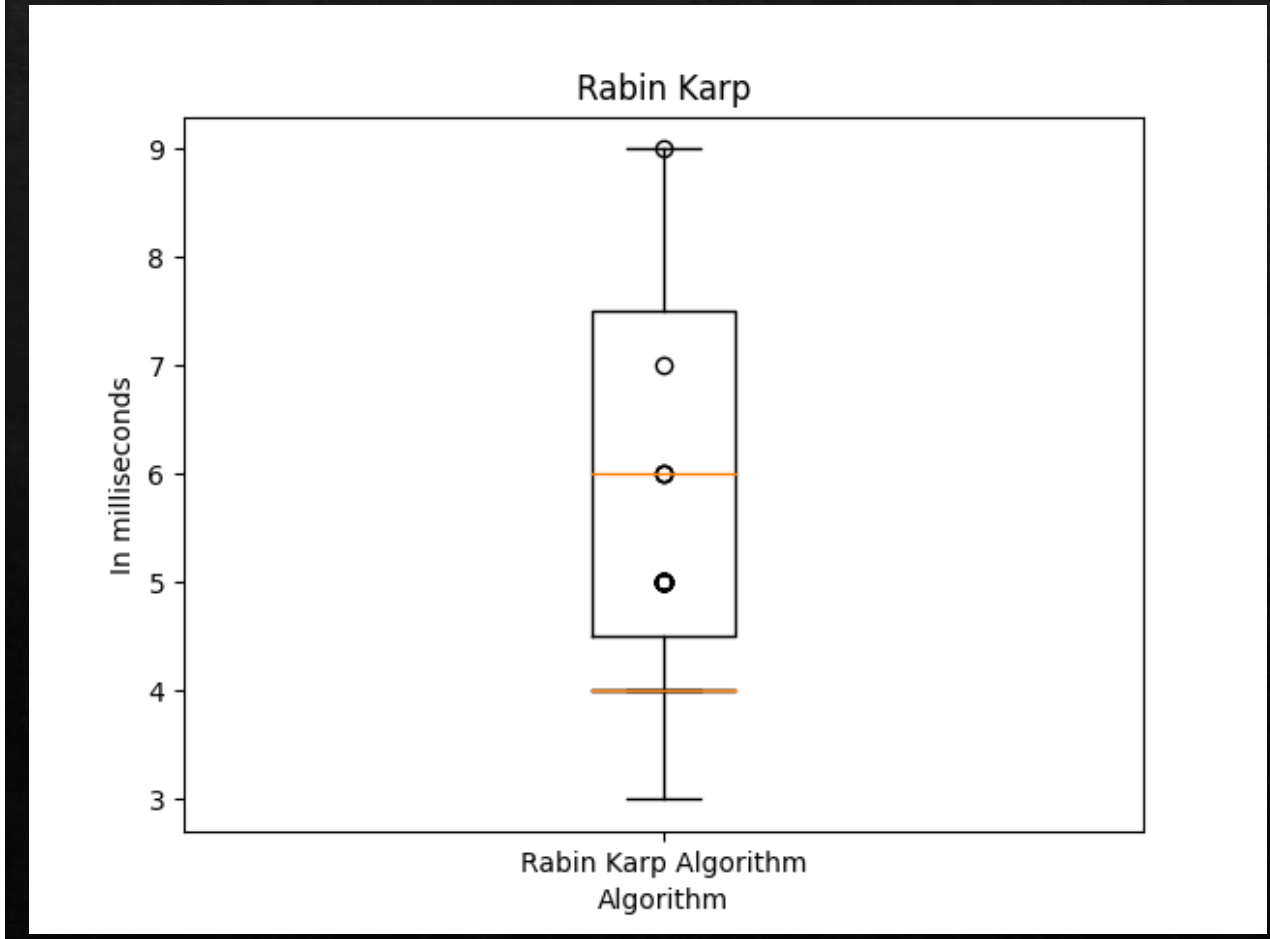
```
void load_jute_book(string& str) {
    // Read the whole file into str.
    load_file("jute-book.txt", str);

    // Extract only the main text of the book, removing the Project Gutenberg
    // header/footer and indices.
    str = str.substr(0x4d7);
}
```

Pattern=the (1000 iterations)

Boyer-Moore-Horspool

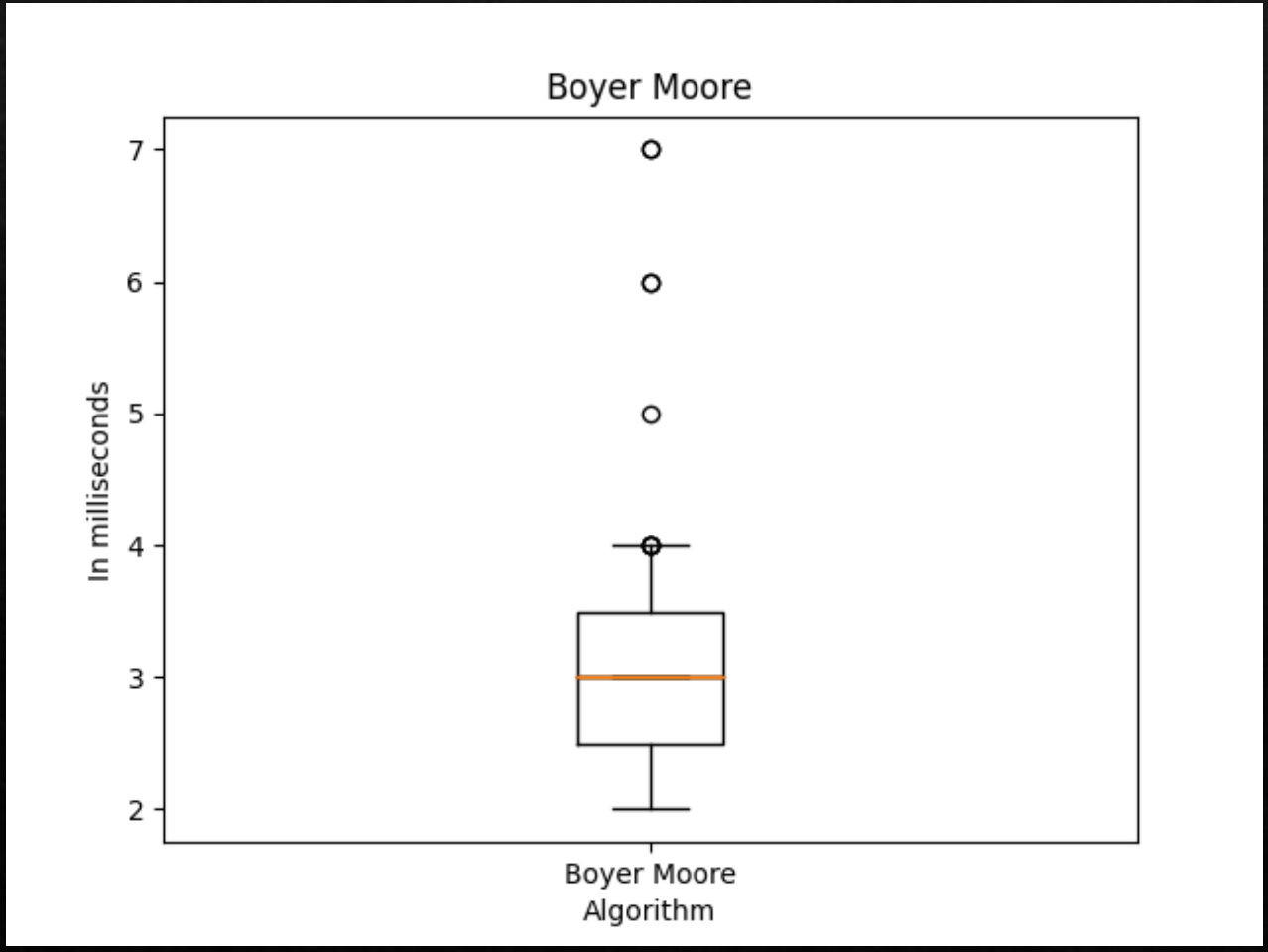


Rabin Karp

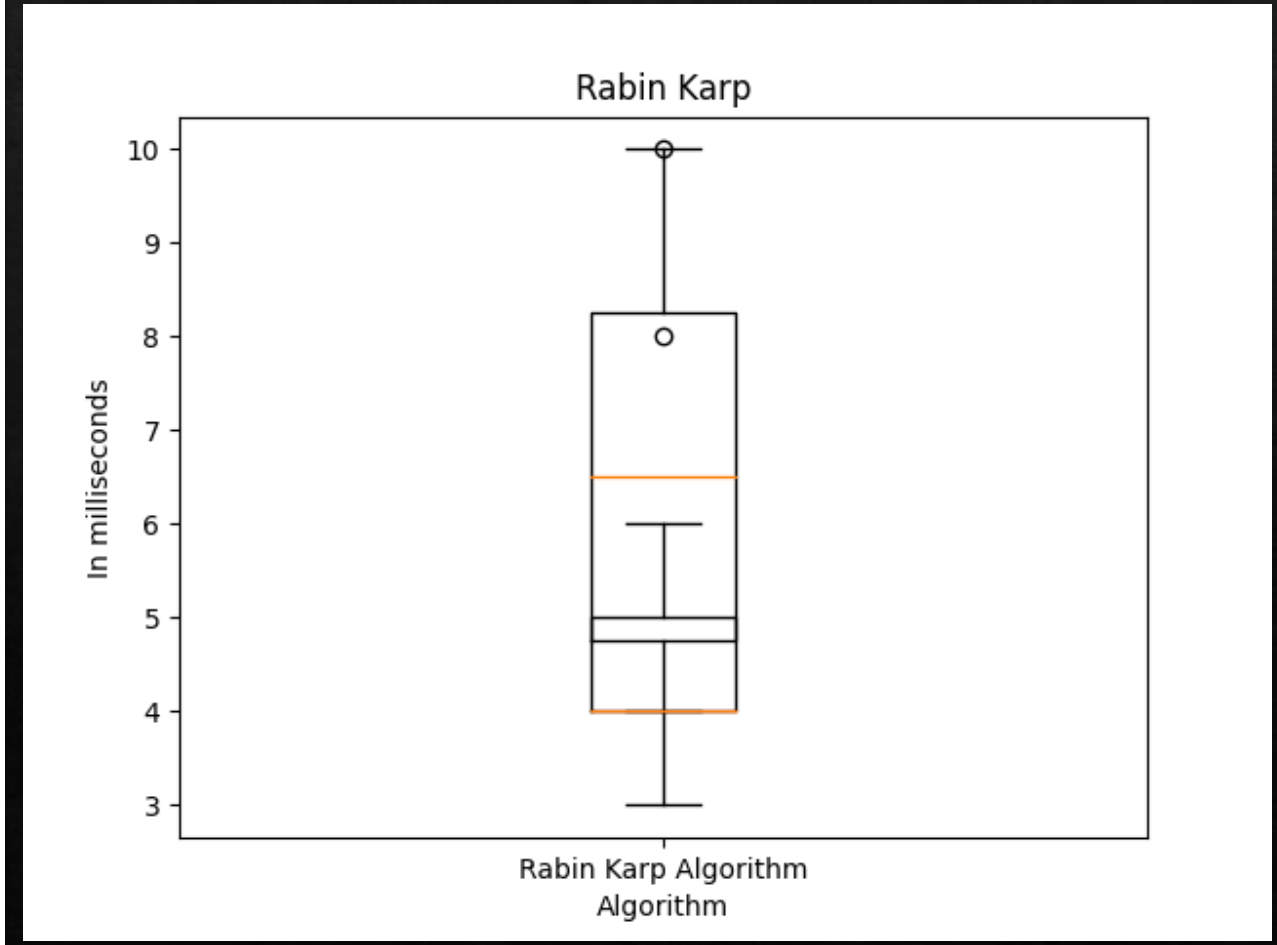


Pattern=terms (1000 iterations)

Boyer-Moore-Horspool

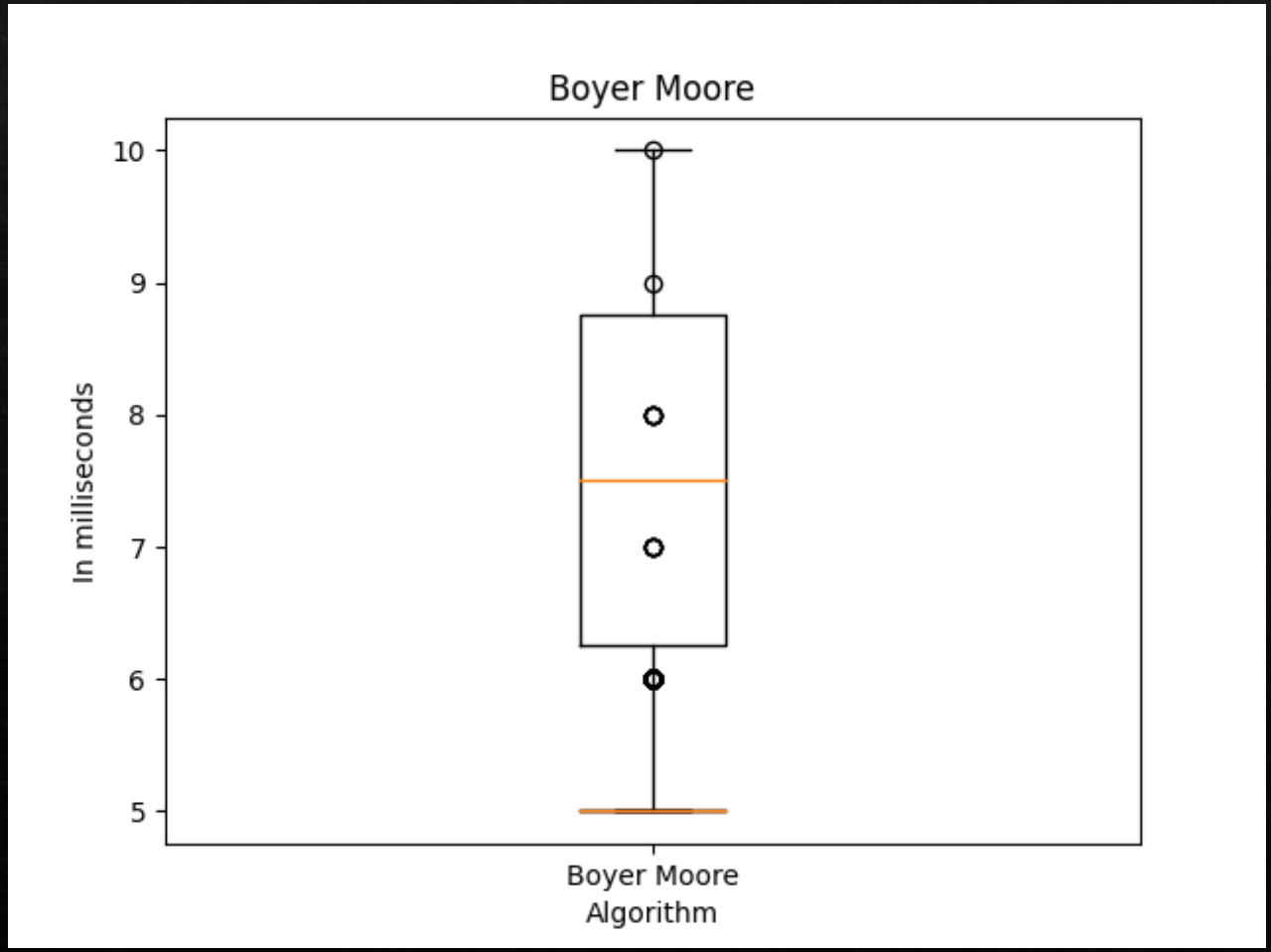


Rabin Karp

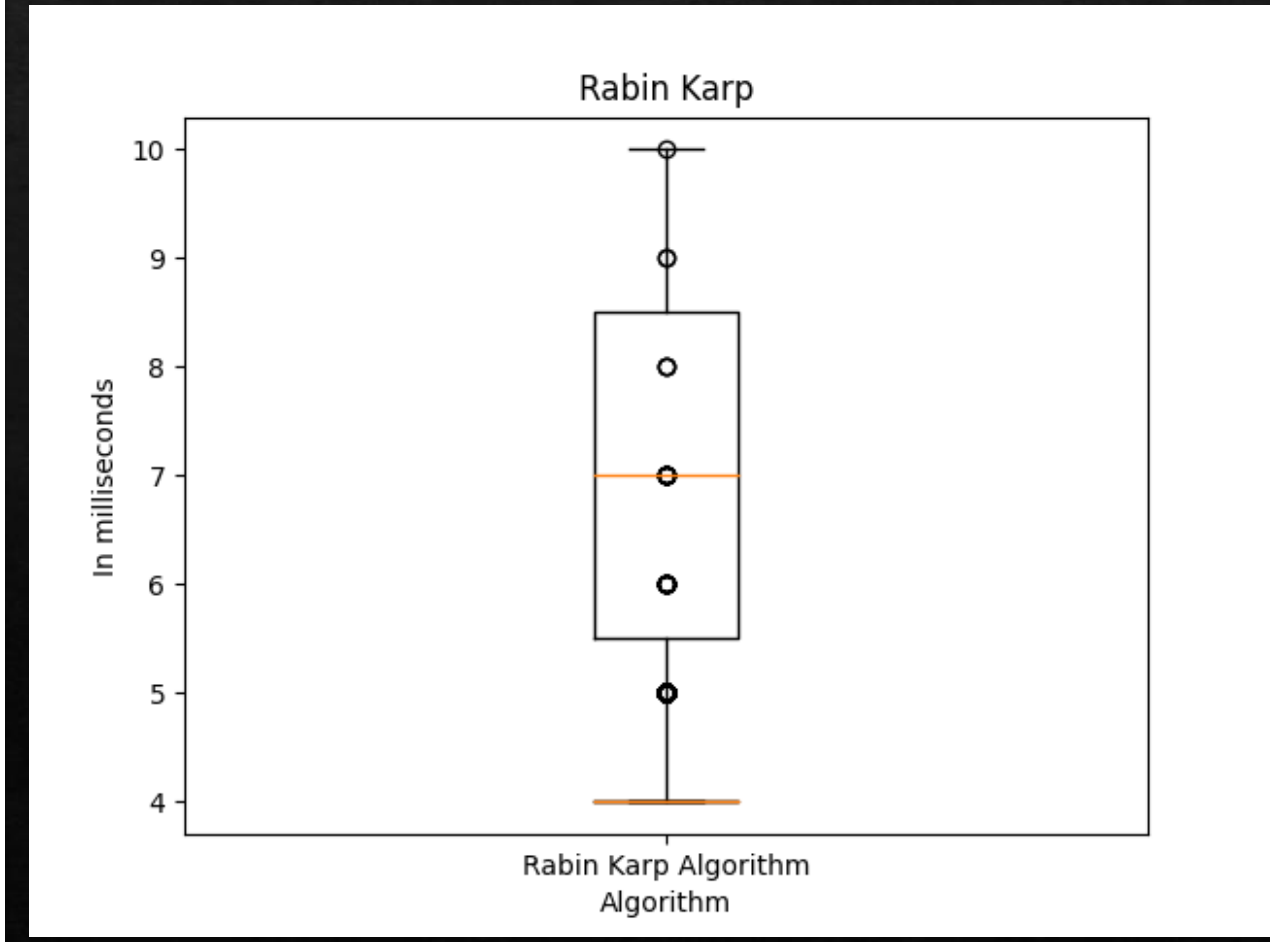


Pattern=the (10000 iterations)

Boyer-Moore-Horspool

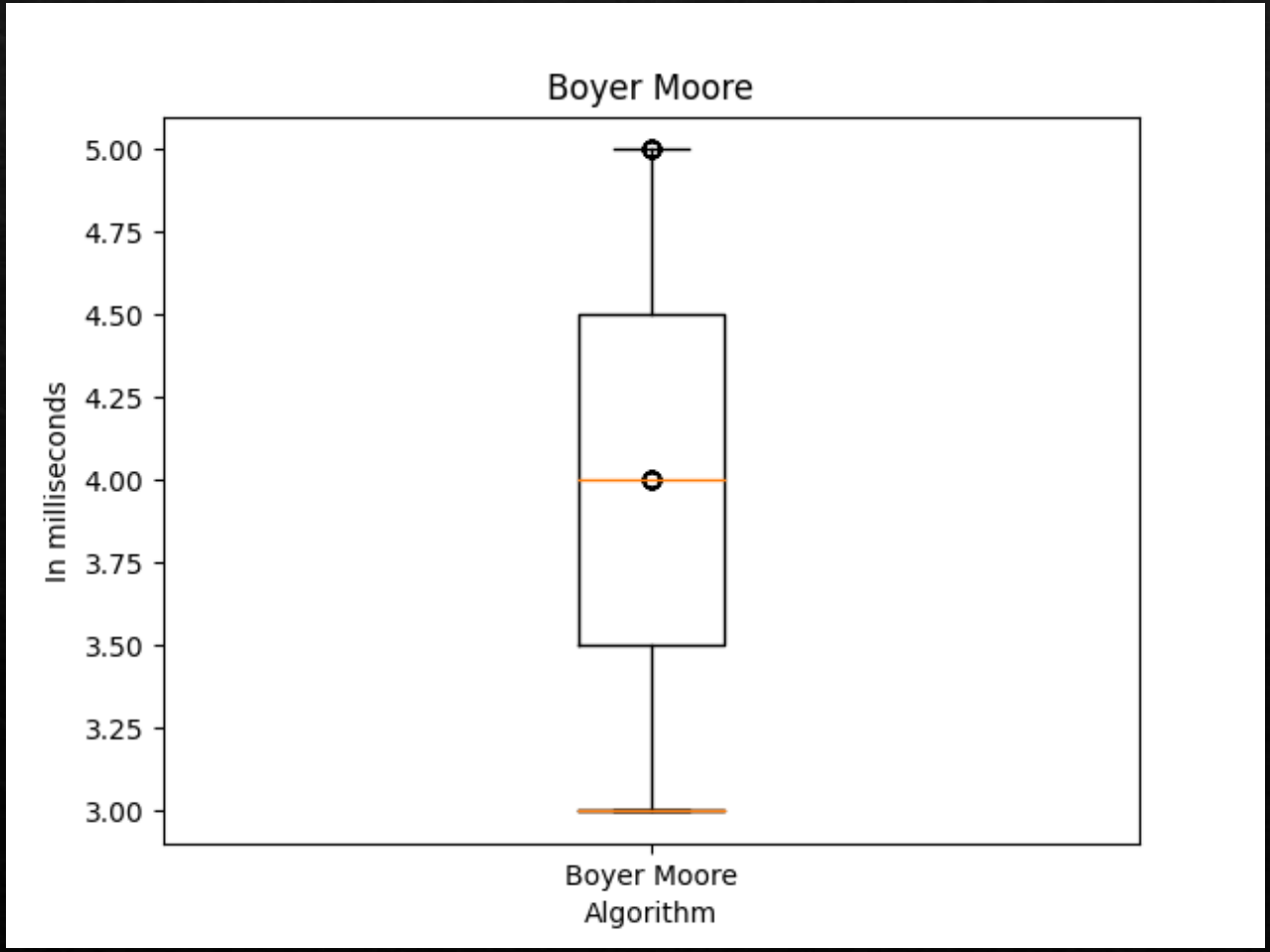


Rabin Karp

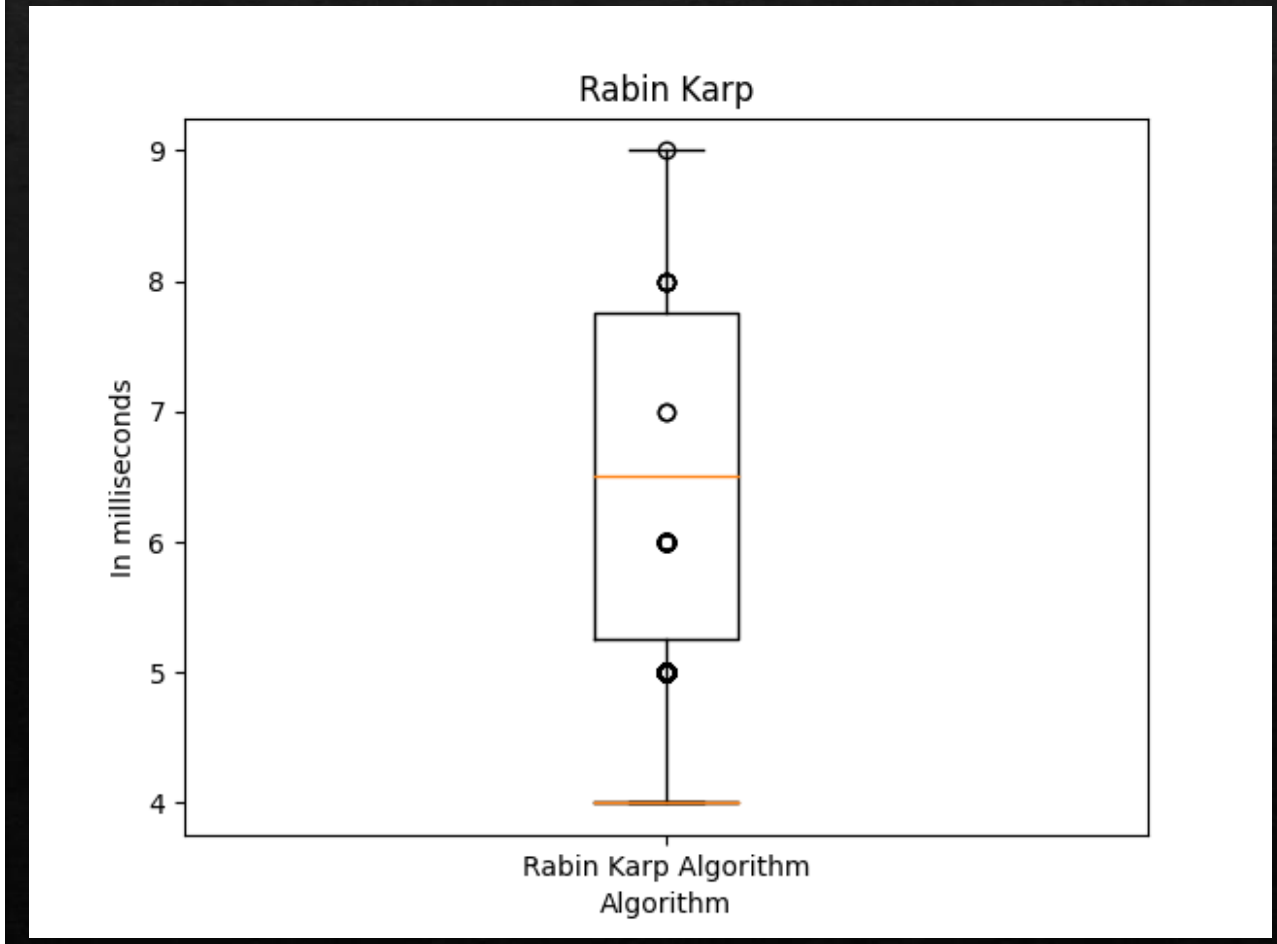


Pattern=terms (10000 iterations)

Boyer-Moore-Horspool



Rabin Karp



Profiling – CPU usage

Searching the

main	100140 (96.95%)	2 (0.00%)	strings.exe	Networking | File...
find_boyer_mo...	54841 (53.10%)	52950 (51.26%)	strings.exe	Networking | Kernel
operator new	1885 (1.83%)	30 (0.03%)	strings.exe	Networking | Kernel
[System Call] nt...	6 (0.01%)	6 (0.01%)	ntoskrnl.exe	Kernel
find_rabin_karp	45276 (43.84%)	9201 (8.91%)	strings.exe	Networking | Kernel
newHash	34017 (32.93%)	17582 (17.02%)	strings.exe	

Searching terms

main	79520 (99.76%)	0 (0.00%)	strings.exe	Networking | File...
find_rabin_karp	46206 (57.97%)	12945 (16.24%)	strings.exe	Kernel
newHash	33243 (41.70%)	16896 (21.20%)	strings.exe	Kernel
operator new	17 (0.02%)	0 (0.00%)	strings.exe	Kernel
[External Call]...	1 (0.00%)	1 (0.00%)	vcruntime140.dll	
find_boyer_mo...	33291 (41.76%)	33267 (41.73%)	strings.exe	Networking | Kernel

Conclusion

Goals

Boyer Moore did get much faster as pattern and text got bigger, meeting time complexity of O(N/M).

Rabin Karp was affected by length of pattern and started to perform worse. However, the Median seemed to almost meet boyer-moore's which is strange if performing much worse.

Box plots

Boyer Moore's boxplots were only squashed together in low iterations. Larger iterations seen timings more spread out, so this varied than expected result.

Rabin Karp's boxplots stayed somewhat the same even though iteration size increased. Max timings always around 9/10 milliseconds.

Time complexity

Boyer-Moore-Horspool: Time complexity matched as expected, O(N/M). As I increased pattern and text size the algorithm only got better as seen.

Rabin Karp: As I didn't implement rolling hash, time complexity will be worse. Instead of O(N), it may be O(N+M). As in my program I subtract first letter and add the next character to get a new hash.

Any questions