Boyer-Moore-Horspool
vs
Rabin-Karp

Motivation

```
bjorn@bjorn:~/logs$ ls -lah
total 8.0K
drwxr-xr-x  2 bjorn bjorn 4.0K Dec 28 11:19 .
drwxr-xr-x 46 bjorn bjorn 4.0K Dec 28 11:24 ..
-rw-r--r--  1 bjorn bjorn  10M Dec 28 11:19 log0.txt
-rw-r--r--  1 bjorn bjorn  10M Dec 28 11:19 log1.txt
-rw-r--r--  1 bjorn bjorn  10M Dec 28 11:19 log2.txt
-rw-r--r--  1 bjorn bjorn  10M Dec 28 11:19 log3.txt
-rw-r--r--  1 bjorn bjorn  10M Dec 28 11:19 log4.txt
-rw-r--r--  1 bjorn bjorn  10M Dec 28 11:19 log5.txt
-rw-r--r--  1 bjorn bjorn  10M Dec 28 11:19 log6.txt
-rw-r--r--  1 bjorn bjorn  10M Dec 28 11:19 log7.txt
-rw-r--r--  1 bjorn bjorn  10M Dec 28 11:19 log8.txt
-rw-r--r--  1 bjorn bjorn  10M Dec 28 11:19 log9.txt
bjorn@bjorn:~/logs$
```

Motivation


```
bjorn@bjorn:~/logs$ ls -lah
4.0K Dec 28 11:19 .
4.0K Dec 28 11:24 ..
 10M Dec 28 11:19 log0.txt
 10M Dec 28 11:19 log1.txt
 10M Dec 28 11:19 log2.txt
 10M Dec 28 11:19 log3.txt
 10M Dec 28 11:19 log4.txt
 10M Dec 28 11:19 log5.txt
 10M Dec 28 11:19 log6.txt
 10M Dec 28 11:19 log7.txt
 10M Dec 28 11:19 log8.txt
 10M Dec 28 11:19 log9.txt
bjorn@bjorn:~/logs$
```

Solution

Pattern to be searched for

Directory with files to be searched

```
bjorn@bjorn:~$ wine BMvsRK.exe "password" "/home/bjorn/logs"
 _  _  __  _  _  _  __
|_) /_\ \/ |_ |_) - |\/| /_\ /_\ |_) |_ - |-| /_\ |_) /`\ |_) /_\ /_\ |__
|_) \_/ |  |__ |\  - |  | \_/ \_/ |\  |__ - |-| \_/ |\  .\_/ |  \_/ \_/ |__
                                                  _
                                                \ / /`\
                                                 v  .\_/
                                  _  _
                                |_) /_\ |_) | |\| - |-/ /_\ |_) |_)
                                | \ /~\ |\  | | \| - | \ /~\ |\  |
                                                                  by Vasco Pinto
/home/bjorn/logs\log2.txt ⇒ Line: 40000 Char: 1 (BM)
/home/bjorn/logs\log8.txt ⇒ Line: 60123 Char: 1 (BM)
/home/bjorn/logs\log2.txt ⇒ Line: 40000 Char: 1 (RK)
/home/bjorn/logs\log8.txt ⇒ Line: 60123 Char: 1 (RK)
Average of Boyer-Moore-Horspool: 797 milliseconds.
Average of Rabin-Karp: 1812 milliseconds.
bjorn@bjorn:~$
```

Files where pattern is present, using Boyer-Moore-Horspool

Files where pattern is present, using Rabin-Karp

Data Structures Used

```
vector<string> getFiles(const string& directory) {

vector<string> files;

//Recursively scan for files
for (const auto& file : filesystem::recursive_directory_iterator(directory)) {

    if (!file.is_directory()) //If it's a directory, don't add it to the vector
        files.push_back(file.path().string()); //Add the file path to the vector
}

return files;
```

Data Structures Used

```
string searchBoyerMooreHorspool(const string& patt, ifstream& file, const char& verbose) {

(…)

unsigned short lookupTable[256];

for (unsigned short i = 0; i < 256; i++)
    lookupTable[i] = (unsigned short)patt.length();
for (unsigned short i = 0; i < patt.length(); i++)
    lookupTable[int(patt[i])] = ((unsigned short)patt.length() - 1) - i;

(…)
```

Performance Characteristics

Algorithm	Average	Best	Worst
Boyer−Moore−Horspool[1]	O(N/M)	O(N/M)	O(NM)
Rabin−Karp[2]	O(N+M)	O(N+M)	O(N+M)

Performance Characteristics

Algorithm	Average	Best	Worst
Boyer−Moore−Horspool[1]	O(N/M)	O(N/M)	O(NM)
Rabin−Karp[2]	O(N+M)	O(N+M)	O(N+M)

1: Quite simple algorithm. Almost all of the time spent in the algorithm is spent on searching for the pattern.

2: More complex algorithm because of hashing. Most of the time spent in the algorithm is related to hashing.

Polynomial Rolling Hash

Hash(“abc”) = int(‘a’) + int(‘b’) + int(‘c’)

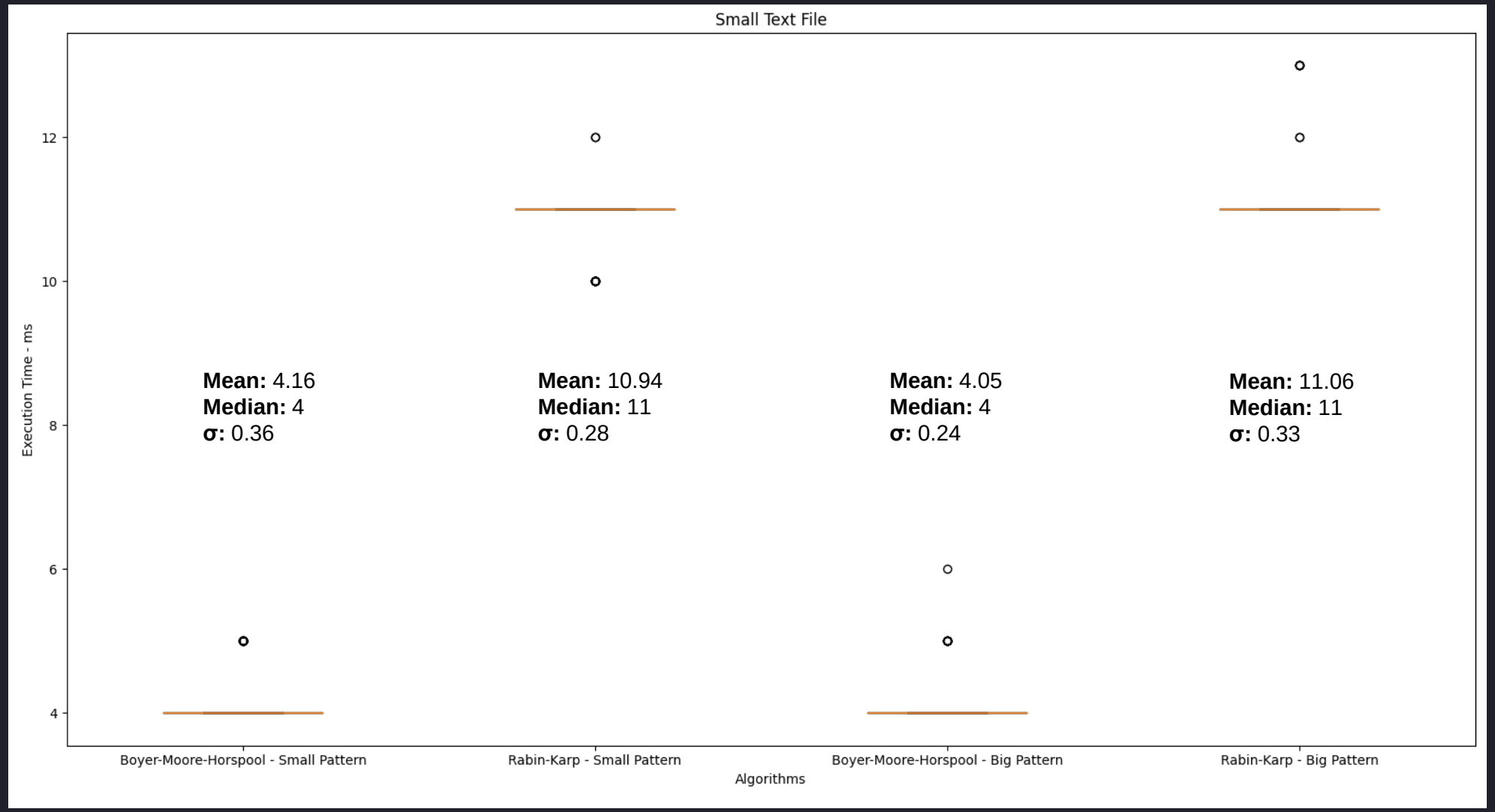
vs

Hash(“abc”) = (int(‘a’) * $base^{m-1}$ + int(‘b’) * $base^{m-2}$ + int(‘c’) * $base^{m-3}$) mod Prime

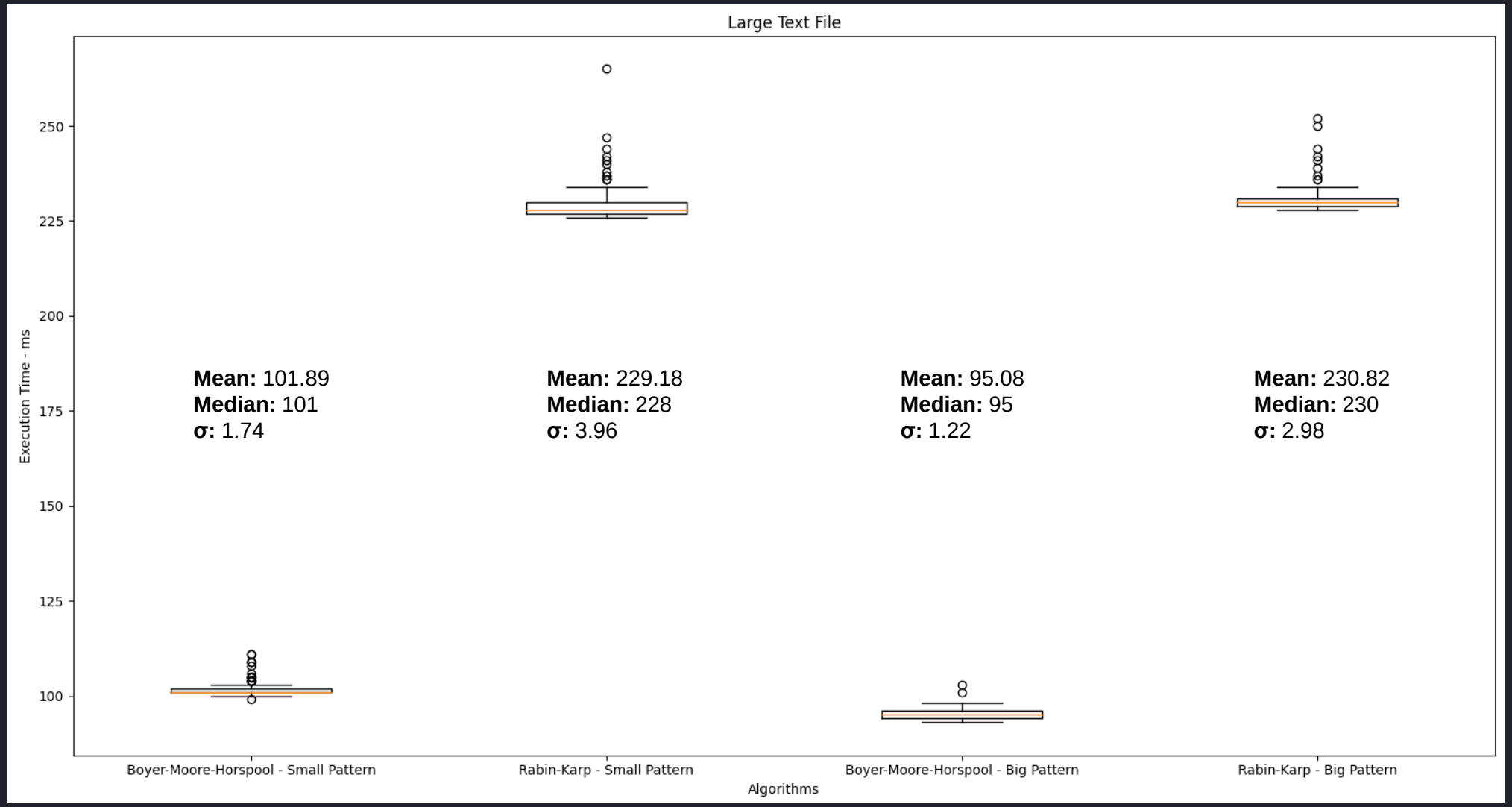
Comparison

Input	Text File	Pattern
Small	741975 Characters 1000 Lines	8 Characters
Big	15128189 Characters 20000 Lines	154 Characters

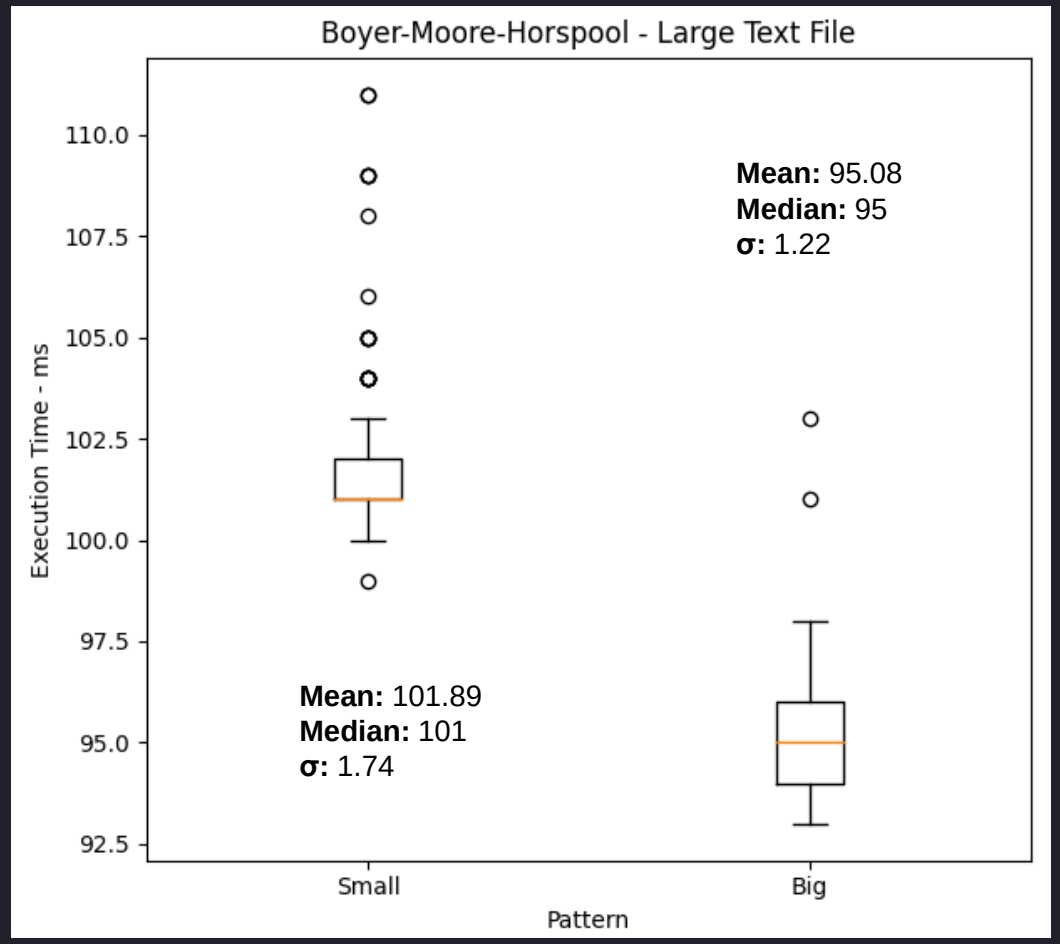
Comparison (x200)

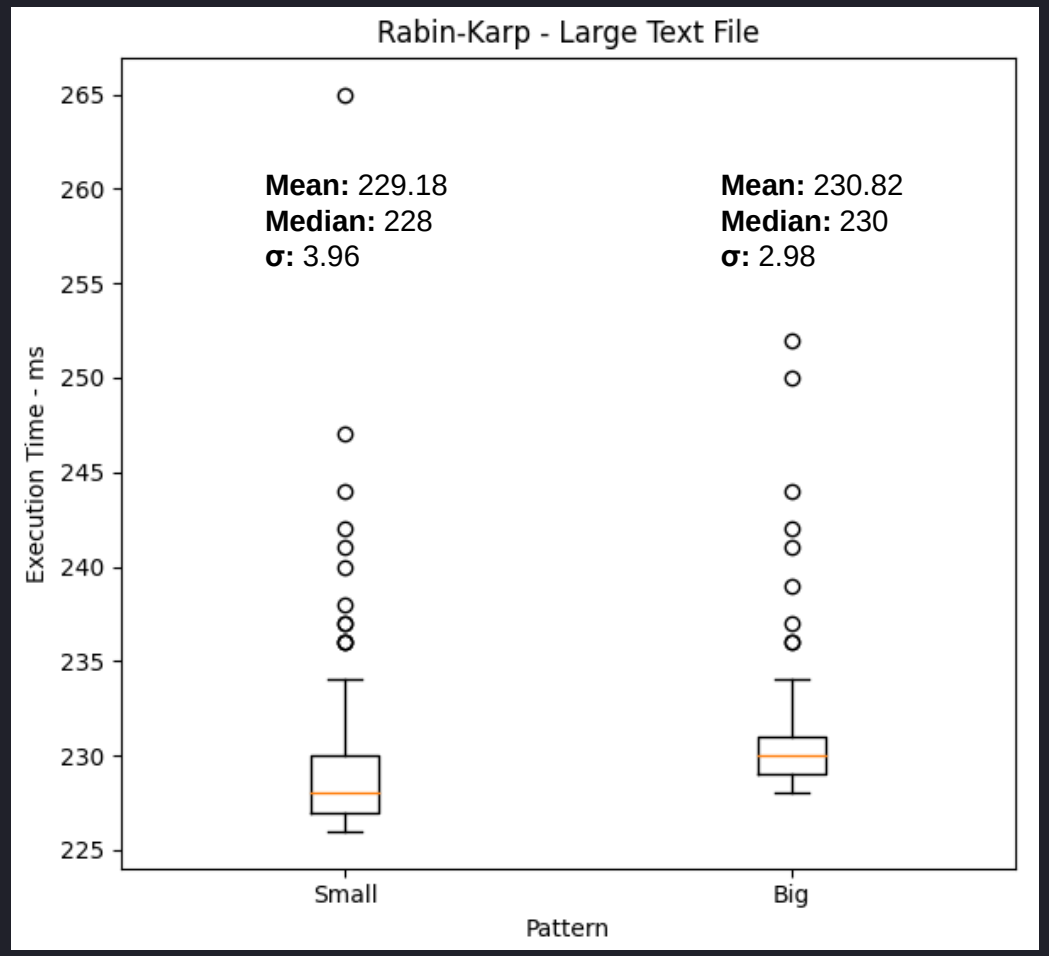


Comparison (x200)



Comparison (x200)





Any Questions?