EMPIRICAL ANALYSIS OF PERFORMANCE OF QUICKSELECT WITH (SEDGEWICK) AND (LOMUTO) PARTITION ALGORITHMS

CPCS223 PROJECT - 2022

STUDENT NAME: ZAINAB ALSAGGAF

ID: 2006531

CONTENTS

INTRODUCTION

Empirical analysis of an algorithm is performed by running a program implementing the algorithm on a sample of input and analyzing the data observed.

It is an alternative to the mathematical analysis of the algorithm's efficiency. and its applicability to any algorithm is the principal strength of this approach.

In this report, we will make an Empirical analysis of the Quickselect algorithm. *Quickselect* is a selection algorithm used to find the kth smallest element in an unordered list (Levitin, 2011). And it is a widely used algorithm.

EXPERIMENT PURPOSE

The purpose of this study is to compare the time efficiency of two approaches of an algorithm that solve the same problem but with different partition methods.

- Quickselect algorithm with Sedgewick partition.
- Quickselect algorithm with Lomuto partition.

ALGORITHMS

Algorithms used in this experiment:

1. Quickselect

Fig21, page 1 (Al-Hashimi M. , 2022)

2. Sedgewick Partition

Fig22, Page 6 (Al-Hashimi D. M., 2022)

3. Lomuto Partition

Introduction to The Design and Analysis of Algorithms Textbook Page 159 (Levitin, 2011)

4. Random Numbers Generator

Pseudocode for generating random numbers:

```
// Fill array with random numbers from 0 to 100
// Input: an integer number with the desired size
// Output: an array filled with random numbers
Algorithm Fill(size)
  a = []
  for  i = 0 and i < size
    a.push(Math.floor(Math.random() * 101))
     i++
  return a
```

5. Find Median Algorithm

```
// Find the median value
// Input: Array
// Output: Median value
function Median(A[0..n-1])
    r = A.length - 1
    l = 0
  return
```

$\left\lfloor l + r - 2 \right\rfloor$

TOOLS

The tools used in this experiment:

- Microsoft Excel
  Recording data in tables, calculating the limits, and drawing graphs.
- Visual Studio Code
  Help in writing and editing the code
- JavaScript Programming language
- Firefox Web browser
  Used developer tools to run and debug the program within the browser.

EFFICIENCY METRIC

An algorithm's time efficiency can be measured in two ways:

1. Basic Operation Count
2. Runtime Count

Basic operation count is done by counting the number of times the algorithm's basic operation is executed. And runtime count is done by counting the time of the program implementing the algorithm.

In this study, we will use both metrics, and after collecting them we will compare between them and choose a preferred one.

PROCEDURE OF IMPLEMENTING THE ALGORITHM

Input Sample

We used an array as an input for both algorithms. The array had the following characteristic:

SAMPLE SIZE

Firstly, we started with a small array size, and then we increased it until we saw a big difference between the two algorithms' efficiencies. The final size we reached was 1,000,000.

SAMPLE RANGE

Array's values were integer values in the range from 0 to 100. The values were generated randomly, since an empirical analysis requires generating random numbers, using a random number generator in the language library, shown in Algorithm 4.

Program Code

The program code generates different input sizes starting from 100,000 until 1,000,000 and loops into the algorithm and makes 10 trials for each input size, as the result may differ for some instances of the same size. For each trial, there's a counter that counts the number of times the basic operation has been executed in the most inner loop and calculated the algorithm's runtime. For this experiment, we chose the kth element to be sorted as always the median value in the array, using Find Median Algorithm (Algorithm 5). Moreover, we made sure to use the same instance for comparing between the two partitions by shuffling the array before using it.

RECORDING DATA

After running the code, all the data have been recorded in tables. Then, calculated the best, average, and worst cases for each input size of the 10 trials. As shown in Tables 1, 2, 3, and 4.

Size	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8	Trial 9	Trial 10	Best	Average	Worst
100000	328127	302812	542683	306074	340850	227151	328960	294025	380265	394283	**227151**	**344523**	**542683**
200000	593430	976699	617686	777146	624749	917600	530420	552360	640721	389398	**389398**	**662020.9**	**976699**
300000	1037024	816138	993992	1026354	818501	1130198	724481	1159855	875073	769394	**724481**	**935101**	**1159855**
400000	1423322	686483	1373128	1012208	995902	1393395	1673664	1416149	1153065	1314808	**686483**	**1244212.4**	**1673664**
500000	998783	1958594	1286542	1547264	1068410	1751334	1277481	1879560	1800178	1724043	**998783**	**1529218.9**	**1958594**
600000	1967220	2250291	1780735	1702781	1507453	1791070	1579235	2573911	1416748	1364566	**1364566**	**1793401**	**2573911**
700000	2301632	2414500	2671346	2041054	2168649	2736749	3142975	2144423	1998592	2328461	**1998592**	**2394838.1**	**3142975**
800000	2285292	2876577	2831881	2854333	1413663	3303043	2608226	2036767	2189663	2948754	**1413663**	**2534819.9**	**3303043**
900000	2088851	1994450	4940028	3113876	2253755	1929679	2239336	2774952	4505119	2573189	**1929679**	**2841323.5**	**4940028**
1000000	3358066	3912236	3623745	3493458	1873785	3304978	2677558	3412468	2843661	4471239	**1873785**	**3297119.4**	**4471239**

Table 1 Sedgewick Basic Operation count

Size	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8	Trial 9	Trial 10	Best	Average	Worst
100000	15	12	13	10	11	7	8	11	13	12	**7**	**11.2**	**15**
200000	9	19	11	13	9	18	9	9	9	9	**9**	**11.5**	**19**
300000	26	17	16	19	17	20	15	20	18	12	**12**	**18**	**26**
400000	48	18	24	21	20	26	26	24	21	25	**18**	**25.3**	**48**
500000	24	33	24	27	36	46	22	36	30	42	**22**	**32**	**46**
600000	46	35	44	35	33	40	35	53	33	37	**33**	**39.1**	**53**
700000	38	78	41	32	36	43	42	35	32	43	**32**	**42**	**78**
800000	41	55	46	45	30	70	40	37	39	45	**30**	**44.8**	**70**
900000	59	44	90	41	62	48	36	38	93	43	**36**	**55.4**	**93**
1000000	59	63	54	53	37	74	47	53	47	58	**37**	**54.5**	**74**

Table 2 Sedgewick Runtime Count

Size	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8	Trial 9	Trial 10	Best	Average	Worst
100000	487759	789758	902310	980367	589584	904434	865491	524188	675294	773424	**487759**	**749260.9**	**980367**
200000	1825695	2216625	1885853	2094967	2262101	1892087	2967666	2059010	2168663	2110791	**1825695**	**2148345.8**	**2967666**
300000	3907599	4645063	3701079	4535518	4020215	4151085	4640594	4634032	4928433	3297005	**3297005**	**4246062.3**	**4928433**
400000	6217023	9406307	7827253	6249519	7969872	8276252	7134592	7436505	7155862	8371841	**6217023**	**7604502.6**	**9406307**
500000	11462853	10782354	10410704	11410394	8880874	9535626	10833284	10170950	9806087	12977985	**8880874**	**10627111.1**	**12977985**
600000	11318499	13570582	17478019	13100632	12535612	15199950	14873795	15032839	14329036	13206521	**11318499**	**14064548.5**	**17478019**
700000	20918622	22104463	16513094	19152642	23483057	17178055	17898575	21457279	22272912	18633695	**16513094**	**19961239.4**	**23483057**
800000	25820752	27404584	26361537	30787289	24964896	20281246	24574535	27095645	25272237	26836685	**20281246**	**25939940.6**	**30787289**
900000	34427871	31446687	33077202	29097139	36318804	36690280	38240058	30710806	31540223	29682079	**29097139**	**33123114.9**	**38240058**
1000000	39200890	40213272	43570092	41788883	44426258	43316892	41988634	43585073	36809315	39891583	**36809315**	**41479089.2**	**44426258**

Table 3 Lomuto Basic Operation Count

Size	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8	Trial 9	Trial 10	Best	Average	Worst
100000	14	23	24	24	10	20	18	11	15	18	**10**	**17.7**	**24**
200000	31	34	29	27	31	26	35	28	26	44	**26**	**31.1**	**44**
300000	69	72	44	57	47	49	47	58	52	40	**40**	**53.5**	**72**
400000	75	188	113	75	111	106	103	87	78	84	**75**	**102**	**188**
500000	152	122	110	138	90	108	124	101	110	146	**90**	**120.1**	**152**
600000	174	191	267	182	188	198	198	184	209	169	**169**	**196**	**267**
700000	277	303	199	234	366	169	193	247	233	243	**169**	**246.4**	**366**
800000	297	317	315	483	266	201	272	306	289	289	**201**	**303.5**	**483**
900000	470	369	413	310	466	444	590	326	386	465	**310**	**423.9**	**590**
1000000	552	488	636	661	663	619	595	544	562	502	**488**	**582.2**	**663**

Table 4 Lomuto Runtime Count

GRAPH THE DATA

Using Excel tools for drawing graphs, we draw a graph for each table as shown in Figures 1, 2, 3, and 4. Drawing a graph can help us identify an algorithm's efficiency class.

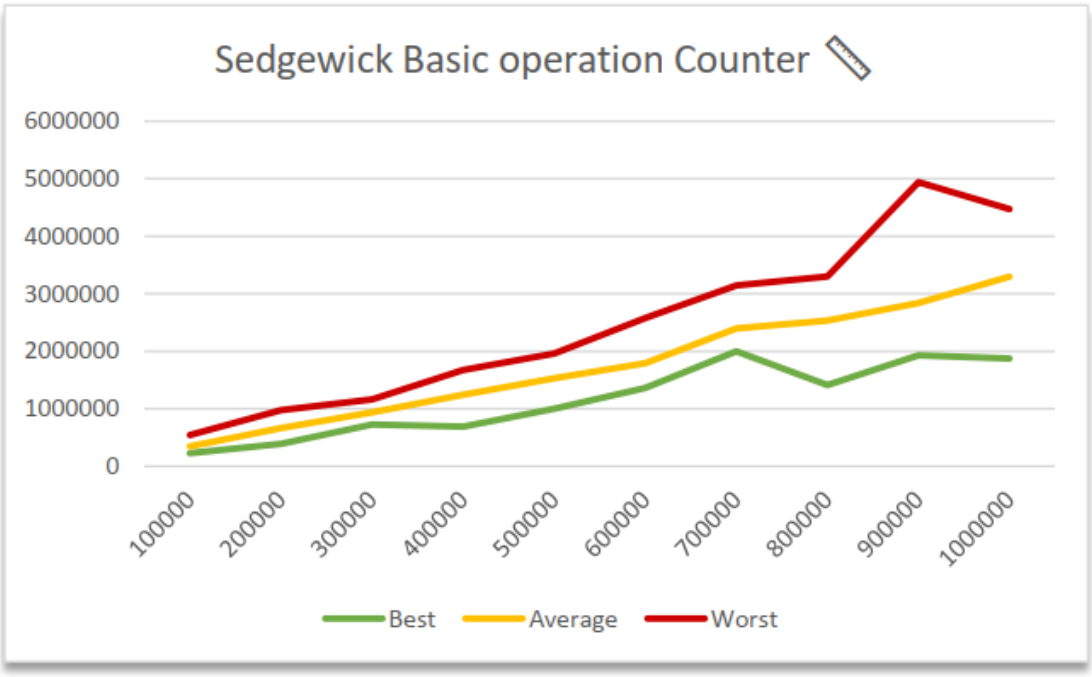

Figure 1 Sedgewick Basic operation Counter

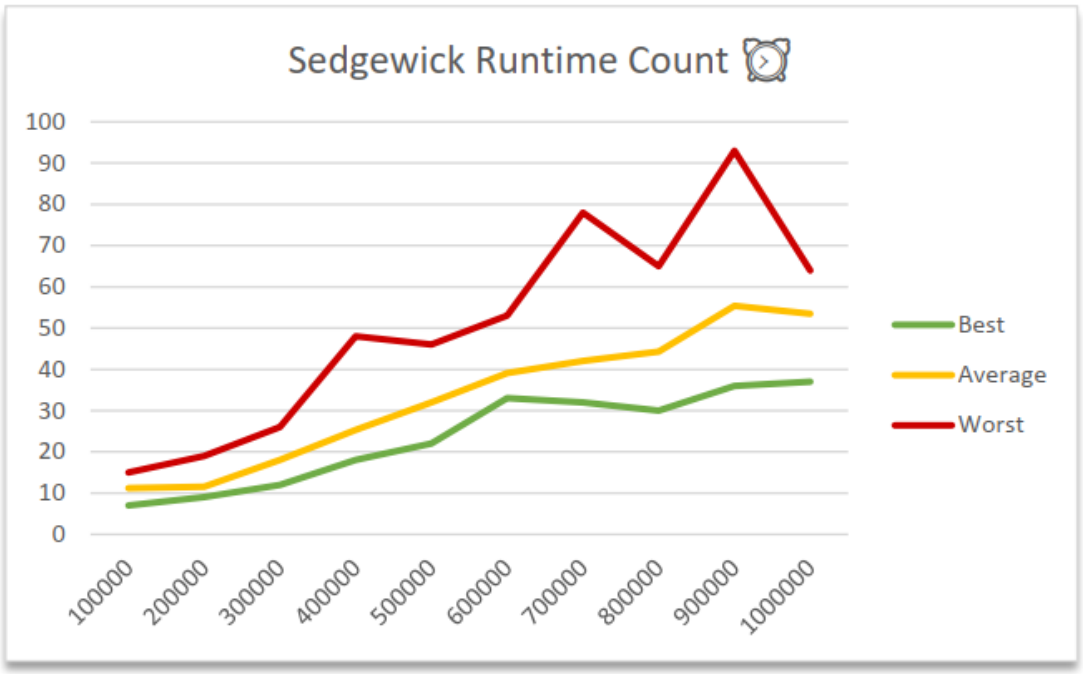

Figure 2 Sedgewick Runtime Count

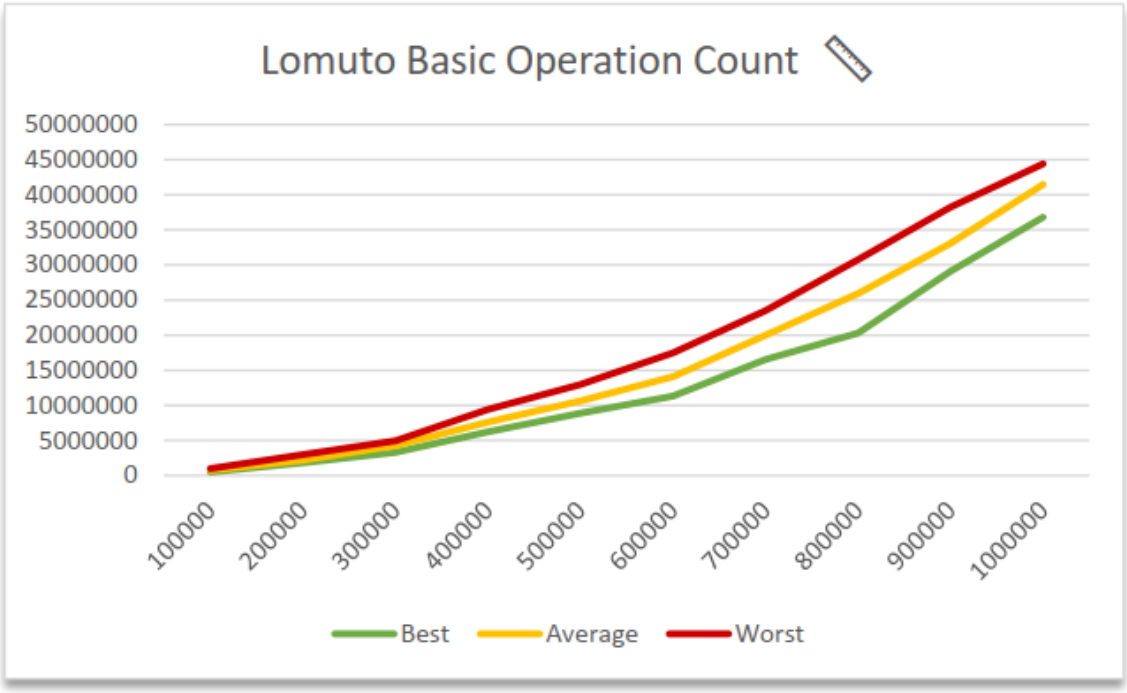

Figure 3 Lomuto Basic Operation Count

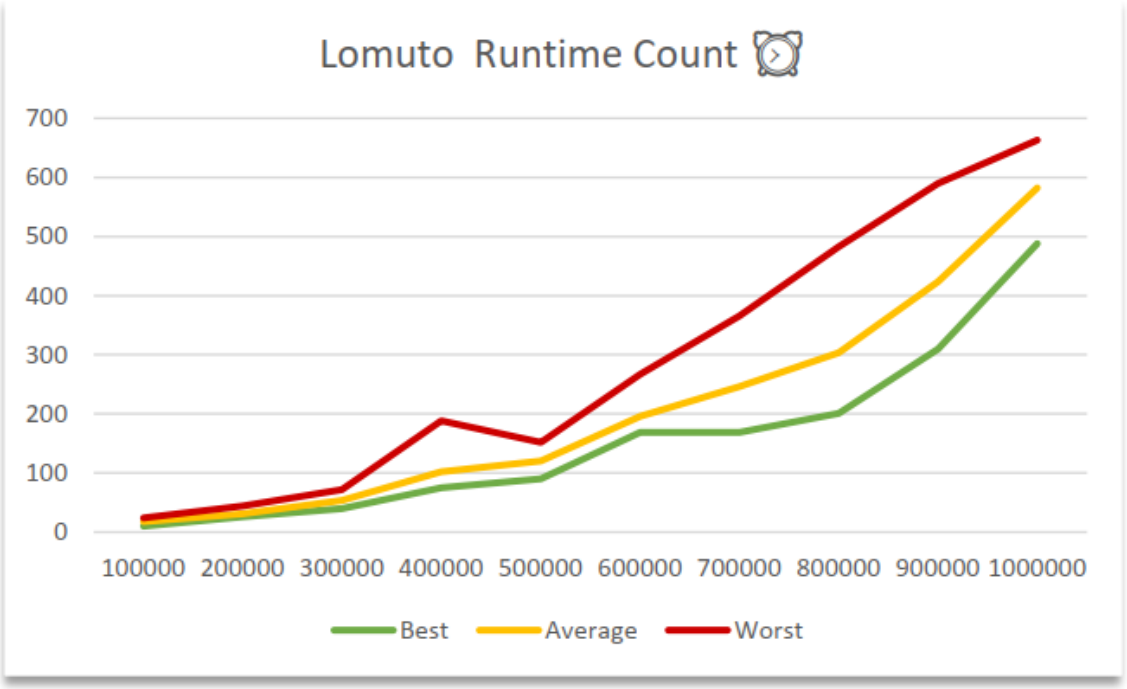

Figure 4 Lomuto Runtime Count

LIMITS

By using Excel tools in calculating limits as n gets large $\lim_{n\to\infty}\left(\frac{t(n)}{g(n)}\right)$

$$Sedgewick\ Basic\ operation\ Counter = \lim_{n\to\infty}\left(\frac{t(n)}{n}\right) = +ve\ Constant$$

$$Sedgewick\ Runtime = \lim_{n\to\infty}\left(\frac{t(n)}{n}\right) = +ve\ Constant$$

$$Lomuto\ Basic\ operation\ Counter = \lim_{n\to\infty}\left(\frac{t(n)}{nlogn}\right) = +ve\ Constant$$

$$Lomuto\ Runtime = \lim_{n\to\infty}\left(\frac{t(n)}{nlogn}\right) = +ve\ Constant$$

ANALYZING DATA

In this experiment, we used two methods for investigating the order of growth, and each method has its unique strength and weakness.

The first one is by drawing a graph, and the second is by using limits as n increases to ∞.

Analyzing the graphs, all four graphs show no variability, which means that we do not have to investigate the best, average, and worst cases.

Graphs 1 and 2 follow the same behavior, they show a straight line, and that leads to efficiency class Θ(n).

Graphs 3 and 4 show a convex shape and their efficiency class is Θ(nlogn).

By using limits to investigate the order of growth, all the limits' results confirm the order of growth we obtained from the graphs. When $\lim_{n\to\infty}\left(\frac{t(n)}{g(n)}\right) = c$ that implies that $t(n)$has the same order of growth of $g(n)$ which means that $t(n) \in \theta\big(g(n)\big)$

PREFERRED METRIC CHOICE

The preferred metric we chose is the basic operation counter.

The physical runtime depends on a specific machine, as we mentioned earlier, Quickselect is a widely used algorithm, and if we used it on a supercomputer the results we obtained will significantly change, unlike the basic operation counter will remain the same.

DISCUSSION OF RESULTS

Comparing to Theory

The mathematical analysis of Quickselect with Lomuto and Sedgewick partitions:

$C_{Best,\ Avg} \in \theta(n)$, $C_{Worst} \in \theta(n^2)$

And by comparing the mathematical with the empirical analysis we can indicate that the empirical analysis is not accurate.

Reasons for Incorrect Analysis

The reason for not getting the right answers is because of the input values range.

In the experiment, we limited our input range by only values starting from 0 to 100. And that was not as random input as we expected.

Another reason is, the number of trials was not enough to investigate the algorithm's best, average, and worst cases.

CONCLUSION

In conclusion, the empirical analysis does not always give an accurate analysis of an algorithm's performance. The weakness of empirical analysis lies in depending on a particle sample of instances, and the computer used in the experiment.

REFERENCES

Al-Hashimi, D. M. (2022, 3 29). *cs223fig22.* Retrieved from CPCS223 Analysis | Design of Algorithms: https://www.hashimi.ws/cs223/figs/cs223fig22.pdf

Al-Hashimi, M. (2022, 3 28). *cs223fig21.* Retrieved from CPCS223 Analysis | Design of Algorithms: https://www.hashimi.ws/cs223/figs/cs223fig21.pdf

Levitin, A. (2011). *Introduction to the Design and Analysis of Algorithms, 3rd Edition.* Pearson.