

Assignment 1

Algorithms and Data Structures

Haider Ali

Student ID: 31371337

FIT2004

Due Date: 20th August, 2021

|  |  |  |
| --- | --- | --- |
| **Base1** | **Time(y1)** | **Time (y2)** |
| log(2) | 37.5257664 | 77.33926 |
| log(4) | 17.6544204 | 42.51335 |
| log(8) | 11.7617509 | 26.38961 |
| log(16) | 9.06538725 | 20.93197 |
| log(32) | 6.98537397 | 16.27453 |
| log(64) | 5.9611609 | 13.66847 |
| log(128) | 5.14704967 | 11.84035 |
| log(256) | 4.72731328 | 11.16113 |
| log(512) | 4.13998365 | 9.811771 |
| log(1024) | 3.71002603 | 8.549147 |
| log(2048) | 3.67416 | 8.179137 |
| log(4096) | 3.52158689 | 7.106082 |
| log(8192) | 3.01896691 | 6.395825 |
| log(16384) | 2.8892405 | 5.995142 |
| log(32768) | 2.83342576 | 5.682808 |
| log(65536) | 2.91222429 | 5.771574 |
| log(131072) | 3.44378495 | 6.222365 |
| log(262144) | 3.98235464 | 7.169834 |
| log(524288) | 5.33773208 | 8.248952 |
| log(1048576) | 8.05746102 | 11.65385 |
| log(2097152) | 13.694391 | 16.49492 |
| log(4194304) | 24.3269718 | 26.6308 |

The complexity of radix sort is O((n+b)log(m)) where n is the number of elements in the list and b is the base and m is the maximum element in the list

From the above graph its clearly visible that if you have a large exponent and many elements in the list, it causes the complexity of the radix sort algorithm to increase. However, if you have a small base, for example 4 or 8, radix sort takes the most time to sort a large set of data. The ideal scenario is when we have a relatively large value of the base (around ) and then, even with a large set of data, we can see radix sort completely sorting the list in around 5 seconds.

Y1 has data in which the values have a range of and Y2 has data in which the values have a range of .

Despite the bases and number of elements being the same for y1 and y2, y2 takes more time to radix sort as the complexity is O((n+b)log(M)) where M is the value of the largest data. Log() > Log() hence the overall time complexity for y2 is greater than y1 as represented by the graph. However when we use a higher base, the graph lines get really close to each other since the base begins to affect the overall value and Log(M) makes a less difference in the value of the complexity.

|  |  |  |
| --- | --- | --- |
| **Base2** | **Time(y3)** | **Time(y4)** |
| 2500000 | 16.23863 | 18.6779366 |
| 3000000 | 19.18771 | 22.0092609 |
| 3500000 | 21.15345 | 25.3454225 |
| 4000000 | 24.07106 | 27.4313116 |
| 4500000 | 26.42248 | 29.1959231 |
| 5000000 | 29.55322 | 31.984762 |
| 5500000 | 33.26661 | 35.0728211 |
| 6000000 | 34.31175 | 36.6628006 |
| 6500000 | 39.14137 | 39.6902978 |

However, if the data set contains large values but the elements only have a few values, it takes more time as shown by the graphs of y3 and y4. The value of base is so much that log(M) in (n+b)log(M) barely makes a difference and instead, the graph is almost linear. The base directly influences the complexity of the radix sort.