Activity 1. Iterative models

C: (nlog(n))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | tLoop1 | tLoop2 | tLoop3 | tLoop4 |
|  | nlog(n) | n2log(n) |  | n3 |
| 100 | 5 \* 10-3 | 158 \* 10-3 | 80 \* 10-2 | 70 \*10-2 |
| 200 | 8 \* 10-3 | 573 \* 10-3 | 341 \* 10-2 | 461 \* 10-2 |
| 400 | 19 \* 10-3 | 2670 \* 10-3 | 1409 \* 10-2 | 3455 \* 10-2 |
| 800 | 45 \* 10-3 | 12539 \* 10-3 | 5891 \* 10-2 | 27053 \* 10-2 |
| 1600 | 114 \* 10-3 | 50470 \* 10-3 | 24971 \* 10-2 | 21806 \* 10-1 |
| 3200 | 221 \* 10-3 | 22276 \* 10-2 | 10458 \* 10-1 | 17125 \* 1 |
| 6400 | 461 \* 10-3 | 8813 \* 10-1 | 46117 \* 10-1 | OoT |
| 12800 | 1020 \* 10-3 | 40168 \* 10-1 | 18550 \* 1 | OoT |
| 25600 | 2176 \* 10-3 | 17326 \* 1 | OoT | OoT |
| 51200 | 4762 \* 10-3 | OoT | OoT | OoT |

To sum up, all the algorithms meet the expected complexity.

Activity 2. Creation of iterative models

|  |  |  |  |
| --- | --- | --- | --- |
| N | tLoop5 | tLoop6 | tLoop7 |
|  | n2log2n | n3log n | n4 |
| 100 | 63 \* 10-2 | 70 \* 1 | 425 \* 1 |
| 200 | 464 \* 10-2 | 557 \* 1 | 6475 \* 1 |
| 400 | 3451 \* 10-2 | 4847 \* 1 | OoT |
| 800 | 29381 \* 10-2 | 40315 \* 1 | OoT |
| 1600 | 21189 \* 10-2 | OoT | OoT |
| 3200 | 19383 \* 1 | OoT | OoT |
| 6400 | OoT | OoT | OoT |
| 12800 | OoT | OoT | OoT |
| 25600 | OoT | OoT | OoT |
| 51200 | OoT | OoT | OoT |

It is plain to see that the numerical results are that of the complexity

Activity 3. Comparison of two algorithms

Comparison of two algorithms with different complexity.

|  |  |  |  |
| --- | --- | --- | --- |
| N | tLoop1 | tLoop2 | t1/t2 |
| 100 | 5 \* 10-3 | 158 \* 10-3 | 0.031646 |
| 200 | 8 \* 10-3 | 573 \* 10-3 | 0.013962 |
| 400 | 19 \* 10-3 | 2670 \* 10-3 | 0.007116 |
| 800 | 45 \* 10-3 | 12539 \* 10-3 | 0.003589 |
| 1600 | 114 \* 10-3 | 50470 \* 10-3 | 0.002259 |
| 3200 | 221 \* 10-3 | 22276 \* 10-2 | 0.992099 |
| 6400 | 461 \* 10-3 | 8813 \* 10-1 | 5.230909 |
| 12800 | 1020 \* 10-3 | 40168 \* 10-1 | 2.539335 |
| 25600 | 2176 \* 10-3 | 17326 \* 1 | 12.55916 |
| 51200 | 4762 \* 10-3 | OoT | - |

A quotient below 1 implies that Loop 1 performs more efficiently than Loop 2. This finding is further supported by their time complexity analysis.

Comparison of two algorithms with the same complexity

|  |  |  |  |
| --- | --- | --- | --- |
| N | tLoop3 | tLoop2 | t1/t2 |
| 100 | 80 \* 10-2 | 158 \* 10-3 | 5.063291 |
| 200 | 341 \* 10-2 | 573 \* 10-3 | 5.951134 |
| 400 | 1409 \* 10-2 | 2670 \* 10-3 | 5.277154 |
| 800 | 5891 \* 10-2 | 12539 \* 10-3 | 4.698142 |
| 1600 | 24971 \* 10-2 | 50470 \* 10-3 | 4.947692 |
| 3200 | 10458 \* 10-1 | 22276 \* 10-2 | 4.694739 |
| 6400 | 46117 \* 10-1 | 8813 \* 10-1 | 0.523284 |
| 12800 | 18550 \* 1 | 40168 \* 10-1 | 0.046181 |
| 25600 | OoT | 17326 \* 1 | - |
| 51200 | OoT | OoT | - |

Because the quotient exceeds 1, it indicates that Loop 2 is more efficient than Loop 3. Despite having the same theoretical complexity, differences in their implementation lead to variations in performance. This happens because both loops follow a logarithmic pattern, but with distinct bases. Consequently, while they are categorized under the same complexity class, their actual execution times differ.

Two Algorithms with the same complexity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | tLoop4 (Python)  tP | tLoop4 (Java  without optimization)  tj1 | tLoop4 (Java  with optimization)  tj2 | tj1/tJ1 | tj2/tj1 |
| 200 | 55 | 461 \* 10-2 | 82 \* 10-3 | 0.083818 | 0.017787 |
| 400 | 383 | 3455 \* 10-2 | 424 \* 10-3 | 0.087247 | 0.012272 |
| 800 | 3262 | 27053 \* 10-2 | 2508 \* 10-3 | 0.081855 | 0.009271 |
| 1600 | 52490 | 21806 \* 10-1 | 17800 \* 10-3 | 0.040741 | 0.008163 |
| 3200 | OoT | 17125 \* 1 | 15341 \* 10-2 | - | 0.008958 |
| 6400 | OoT | OoT | 8761 \* 10-1 | - | - |

Analyzing the quotients reveals that t43 (optimized) is more efficient than t42 (unoptimized), which in turn surpasses t41 (Python). Although they share the same theoretical complexity, differences in performance arise due to Java's optimizations. These optimizations become particularly beneficial when dealing with large input sizes, resulting in reduced execution times.