Activity 1. Some iterative models

Table 1(times in milliseconds and Without Optimization)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | tLoop1(ms) | tLoop2(ms) | tLoop3(ms) | tLoop4(ms) |
| 100 | 68/10^4=0,0068 | 165/10^3=0,165 | 84/10^2=0,82 | 68/10^2=0,68 |
| 200 | 120/10^4=0,0120 | 575/10^3=0,575 | 353/10^2=3,53 | 476/10^2=4,76 |
| 400 | 316/10^4=0,0316 | 2648/10^3=2,648 | 1441/10^2=14,41 | 3613/10^2=36,13 |
| 800 | 661/10^4=0,0661 | 12128/10^3=12,128 | 6106/10^2=61,06 | 26914/10^2=269,14 |
| 1600 | 1592/10^4=0,1592 | 48082/10^3=48,082 | 25641/10^2=256,41 | 2107 |
| 3200 | 3256/10^4=0,3256 | 2175/10=217,5 | 10772/10=1077,2 | 16860 |
| 6400 | 7120/10^4=0,7120 | 8707/10=870,7 | 45943/10=4594,5 | OoT |
| 12800 | 14879/10^4=1,4879 | 39059/10=3905,9 | 19077 | OoT |
| 25600 | 32444/10^4=3,2444 | 17349/1=17349 | OoT | OoT |
| 51200 | 61066/10^4=6,1066 | OoT | OoT | OoT |

The loop1 has a complexity of O(log(n)\*n) we can see the results of the timings increase like that (more or less). The loop2 has a complexity of O(log(n)\*n^2) much larger than loop1 and we can see that in the time measurements. Loop3’s complexity is O(n^2\*log(n)), that is the same as the loop2 and so we can see that the time measurements are similar, they are not equals because this complexity are theoretical. The complexity of loop4 O(n^3) that is why it grows faster than the rest.

Activity 2. Creation of iterative models of a given time complexity

Table 2(time in milliseconds and WITHOUT OPTIMIZATION)

|  |  |  |  |
| --- | --- | --- | --- |
| N | tLoop5(ms) | tLoop6(ms) | tLoop7(ms) |
| 100 | 44/10=4,4 | 55 | 555 |
| 200 | 197/10=19,7 | 449 | 8785 |
| 400 | 925/10=92,5 | 3856 | OoT |
| 800 | 4333/10=433,3 | 33662 | OoT |
| 1600 | 2022 | OoT | OoT |
| 3200 | 9207 | OoT | OoT |
| 6400 | 41948 | OoT | OoT |

The results are ass expected being loop5 the fastest of them and loop7 the slowest due to their complexities.

Activity 3. Two algorithms with different complex.

Table3 (times in milliseconds and WITHOUT OPTIMIZATION)

|  |  |  |  |
| --- | --- | --- | --- |
| N | tLoop1(ms) | tLoop2(ms) | T1 / T2 |
| 100 | 68/10^4=0,0068 | 165/10^3=0,165 | 0,041212 |
| 200 | 120/10^4=0,0120 | 575/10^3=0,575 | 0,02087 |
| 400 | 316/10^4=0,0316 | 2648/10^3=2,648 | 0,011934 |
| 800 | 661/10^4=0,0661 | 12128/10^3=12,128 | 0,00545 |
| 1600 | 1592/10^4=0,1592 | 48082/10^3=48,082 | 0,003311 |
| 3200 | 3256/10^4=0,3256 | 2175/10=217,5 | 0,001497 |
| 6400 | 7120/10^4=0,7120 | 8707/10=870,7 | 0,000818 |
| 12800 | 14879/10^4=1,4879 | 39059/10=3905,9 | 0,000381 |
| 25600 | 32444/10^4=3,2444 | 17349/1=17349 | 0,000187 |
| 51200 | 61066/10^4=6,1066 | OoT | … |

Activity 4. Two algorithms with same complex.

|  |  |  |  |
| --- | --- | --- | --- |
| N | tLoop3(ms) | tLoop2(ms) | T1 / T2 |
| 100 | 84/10^2=0,82 | 165/10^3=0,165 | 4,969697 |
| 200 | 353/10^2=3,53 | 575/10^3=0,575 | 6,13913 |
| 400 | 1441/10^2=14,41 | 2648/10^3=2,648 | 5,441843 |
| 800 | 6106/10^2=61,06 | 12128/10^3=12,128 | 5,034631 |
| 1600 | 25641/10^2=256,41 | 48082/10^3=48,082 | 5,332765 |
| 3200 | 10772/10=1077,2 | 2175/10=217,5 | 4,952644 |
| 6400 | 45943/10=4594,5 | 8707/10=870,7 | 5,276789 |
| 12800 | 19077 | 39059/10=3905,9 | 4,88415 |
| 25600 | OoT | 17349/1=17349 | … |
| 51200 | OoT | OoT | … |

Activity 5. Same algorithm different environments

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Python-t41 | Java without optimization-t42 | Java with optimization-t43 | T42/T41 | T43/T42 |
| 200 | 357 / 10 = 35,7 | 80 / 10 = 8 | 76 / 100 = 0,78 | 0,22409 | 0,0975 |
| 400 | 2813 / 10 = 281,3 | 61 | 558 / 100 = 5,58 | 0,21685 | 0,091475 |
| 800 | 23103 / 10 = 2310,3 | 474 | 4275 / 100 = 42,75 | 0,205168 | 0,09019 |
| 1600 | 19938 | 3784 | 3361 / 10 = 336,1 | 0,189788 | 0,088821 |
| 3200 | OoT | 30191 | 26639 / 10 = 2663,9 | OoT | 0,088235 |
| 6400 | OoT | OoT | 21180 ms | OoT | OoT |