Activity 1. Bubble algorithm

|  |  |  |  |
| --- | --- | --- | --- |
| n (10000) | t ordered | t reversed | t random |
| 1 | 319 | 1512 | 1072 |
| 2 | 1271 | 5962 | 4252 |
| 4 | 5064 | 23892 | 17041 |
| 8 | 20172 | 96432 | 67629 |
| 16 | 80878 | OoT | OoT |

t ordered follows a quadratic O(n2) time as each time you double the n, the time increases in 22

t reversed follows also a quadratic O(n2) time

t random follows also a quadratic time

The 3 follows a quadratic time as the bubble algorithm is O(n2) for sorted, unsorted, average,…

Activity 2. Selection algorithm

|  |  |  |  |
| --- | --- | --- | --- |
| n (10000) | t ordered | t reversed | t random |
| 1 | 319 | 283 | 310 |
| 2 | 1249 | 1124 | 1215 |
| 4 | 4995 | 4486 | 4816 |
| 8 | 19915 | 17867 | 19242 |
| 16 | 78723 | 71469 | 77405 |

t ordered follows a quadratic O(n2) time

t reversed follows also a quadratic O(n2) time

t random follows also a quadratic time

The 3 follows a quadratic time as the selection algorithm is O(n2) as the bubble, for sorted, unsorted, average,…

Activity 3. Insertion algorithm

|  |  |  |  |
| --- | --- | --- | --- |
| n (10000) | t ordered | t reversed | t random |
| 1 | LoR | 295 | 153 |
| 2 | LoR | 1161 | 580 |
| 4 | LoR | 4625 | 2341 |
| 8 | LoR | 18550 | 9300 |
| 16 | LoR | 75583 | 37311 |
| 32 | LoR | OoT | OoT |
| 64 | LoR | OoT | OoT |
| 128 | LoR | OoT | OoT |
| 256 | LoR | OoT | OoT |
| 512 | 91 | OoT | OoT |
| 1024 | 182 | OoT | OoT |
| 1024 | 361 | OoT | OoT |
| 2048 | 723 | OoT | OoT |
| 4096 | 1442 | OoT | OoT |
| 8192 | 2872 | OoT | OoT |

t ordered is so fast as this algorithm checks if it is sorted in each iteration so if it is sorted in x iterations it stops

t reversed and t random are as normal with this algorithm O(n2)

Activity 4. Quicksort algorithm

|  |  |  |  |
| --- | --- | --- | --- |
| n (250000) | t ordered | t reversed | t random |
| 1 | LoR | 94 | 95 |
| 2 | 61 | 189 | 189 |
| 4 | 125 | 402 | 403 |
| 8 | 256 | 867 | 867 |
| 16 | 534 | 1886 | 1872 |
| 32 | 1088 | 4218 | 4226 |
| 64 | 2245 | 10276 | 10276 |

t ordered is faster than the quicksort as it does not do that much comparisons so it is O(nlogn), the rest as we chose a good pivot are also O(nlogn) for reversed and random.

Time for O(n2) algorithms:

N2 = k \* N1 -> k = N2/N1 -> (N2^2/N1^2) = k^2

T2 = k^2 \* T1

k = (160 \* 10^5 / 8 \* 10^5) = 20

86400000 miliseconds is a year

**Bubble:**

N = 8 \* 10^5 -> t = 67629

N = 1.6 \* 10^7 -> t = 20^2 \* 67629 = 27051600

In days = 0.31309722222

**Selection:**

N = 8 \* 10^5 -> t = 19242

N = 1.6 \* 10^7 -> t = 20^2 \* 19242 = 7696800

In days = 0.08908333333

**Insertion:**

N = 8 \* 10^5 -> t = 9300

N = 1.6 \* 10^7 -> t = 20^2 \* 9300 = 3720000

In days = 0.04305555555

Activity 5. Quicksort + Insertion algorithm

|  |  |
| --- | --- |
| n (16 \* 10^6) | t random |
| Quicksort | 11018 |
| Quicksort+Insertion (k=5) | 10770 |
| Quicksort+Insertion (k=10) | 10596 |
| Quicksort+Insertion (k=20) | 10411 |
| Quicksort+Insertion (k=30) | 10167 |
| Quicksort+Insertion (k=50) | 9677 |
| Quicksort+Insertion (k=100) | 8197 |
| Quicksort+Insertion (k=200) | 5338 |
| Quicksort+Insertion (k=500) | 4613 |
| Quicksort+Insertion (k=1000) | 4378 |

The thing is that for really large values quicksort is the best algorithm, and for low values the insertion algorithm gets faster and faster, so until a limit (I don’t know it), the combination of both algorithms will be faster than only the quicksort algorithm alone