|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1000 | 10,000 | 30,000 | Sorted 30k | Reversed 30k |
| Selection | 0.0369400.. | 3.4567954.. | 30.9941.. | 41.48012.. | 43.2729325.. |
| Insert | 0.0478754.. | 5.1163210.. | 47.4915.. | 0.0089859.. | 125.754907.. |
| Merge | 0.006990.. | 0.3760135.. | 3.08679.. | 4.2976417.. | 4.32039260.. |

2.

Selection: complexiteit van O(n2). Je eerste for loop in het algoritme zorgt al voor een complexiteit van O(n), omdat je minimaal 1x door de hele lijst heen moet. De for loop binnen de eerste for loop zorgt dan voor een complexiteit van O(n2). Worst case is in dit geval de reversed gesorteerde lijst, want bij deze doet het algoritme het langst over. De average case is de gesorteerde lijst. De best case is de willekeurig gesorteerde lijst, hier doet het algoritme niet zo lang over.

Insert: complexiteit van O(n2). Je eerste for loop in het algoritme zorgt al voor een complexiteit van O(n), omdat je minimaal 1x door de hele lijst heen moet. De while loop binnen de eerste for loop zorgt dan voor een complexiteit van O(n2), want in het ergste geval moet je weer door de hele lijst heen. De worst case in dit geval is de reversed gesorteerde lijst. Hier doet het algoritme er aanzienlijk lang over. De best case is de gesorteerde lijst, hier is het algoritme namelijk heel snel klaar mee. De average case is dus de willekeurig gesorteerde lijst.

Merge: complexiteit van O (n log n). Als je een lijst van 8 willekeurige getallen hebt, dan moet je 3 keer splitsen om alle getallen los te krijgen en daarna moet je nog 8 keer mergen. Stel je hebt 16 getallen (dus twee keer zo veel) dan moet je 4 keer splitsen om alle getallen los te krijgen. Daarna moet je nog 16 keer mergen. Een lijst moet twee keer zo groot worden om 1 stap toe te voegen. Dit is dus log n. Vervolgens moet er nog n keer worden gemerged. Dus O(n log n).

De best case voor een merge sort is een lijst met getallen waarbij hij zo min mogelijk getallen met elkaar moet vergelijken. De worst case is een lijst met getallen waarbij hij zo vaak mogelijk getallen met elkaar moet vergelijken. De average case is dan een lijst waarbij je niet per se zo vaak of zo min mogelijk getallen moet vergelijken. In mijn geval is de gesorteerde lijst de average case, de gesorteerde reversed lijst de worst case (echter het verschil tussen de twee qua tijd is niet erg groot) en de best case is de willekeurig gesorteerde lijst.

3. qua tijd complexiteit zijn de iteratieve en recursieve versie hetzelfde, want het blijft hetzelfde algoritme wat nog dezelfde stappen zet. Echter qua ruimte complexiteit zijn ze ongelijk. Een iteratieve versie kan een lijst van 1000 elementen sorteren. Bij de recursieve versie is er een grote kans dat deze geen 1000 elementen kan sorteren wegens een stack overflow.