Datastructures

Opdracht 2 – High scores

# 

|  |  |
| --- | --- |
| Naam (studentnummer) | Daan Befort (500727616) |
| Naam (studentnummer) | Kevin Huijzendveld (….) |
| Datum | 11/10/18 |

Inhoudsopgave

[Inleiding 3](#_Toc527039399)

[Insertion-sort implementeren 4](#_Toc527039400)

[Grafiek 4](#_Toc527039401)

[Bucket-sort implementeren 5](#_Toc527039402)

[Grafiek 6](#_Toc527039403)

[Priority queue 7](#_Toc527039404)

[Conclusie 8](#_Toc527039405)

# Inleiding

In dit document leggen we uit hoe we insertion-, bucket-sort en priority queue hebben geïmplementeerd. #TODO kan jij dit beter maken?

# Insertion-sort implementeren

We hebben als eerste de insertion-sort geïmplementeerd. Tijdens het toevoegen van een highscore wordt er ook meteen gesorteerd. De insertion-sort werk als volgt:

**private void** insertionSort(){  
 **for** (**int** i = 1; i < **highScores**.size(); i++) {  
 Player currentPlayer = **highScores**.get(i);  
 **int** current = i;  
 **while** (current > 0 && **highScores**.get(current - 1).getHighScore() < currentPlayer.getHighScore()) {  
 **highScores**.set(current, **highScores**.get(current-1));  
 current--;  
 }  
 **highScores**.set(current, currentPlayer);  
 }  
}

We gaan door de lijst met alle highscorers en kijken per highscore of degene die voor de highscore in de lijst staat kleiner is. Zo ja, dan verwisselen we de twee highscores. Hier gaan we mee door tot dat de highscore voor de highscore niet kleiner is. Zo niet gaan we naar de volgende highscore in de lijst. Dit checken we tot en met de laatste highscore in de lijst. Wanneer dit afgelopen is is de lijst gesorteerd met de *insertion-sort* methode.

## Grafiek

Hieronder vind je de grafiek die bij insertion-sort hoort. We zien hierin hoelang het duurt om het sorteren met insertion-sort. We beginnen met 100 highscores en verdubbelen de waarde steeds. De maximaal aantal highscores is 10000000. Maar wanneer het sorteren langer dan 15 seconden duurt gaan we niet verder.

# Bucket-sort implementeren

We hebben als tweede bucket-sort geimplementeerd. Wanneeer er een highscore wordt toegevoegd gaan we meteen sorteren met de bucket-sort. Dit werkt als volgt:

**public** BucketSortHighScores() {  
 **int** amountOfBuckets = (***MAX\_HIGH\_SCORE*** / ***MAX\_ARRAY\_SIZE***) + 1;  
 **for** (**int** i = 0; i < amountOfBuckets; i++) {  
 **buckets**.add(**new** ArrayList<>());  
 }  
}

Bij het gebruiken van de bucket-sort worden er buckets aangemaakt. Dit zijn emmers waarin highscores kunnen worden toegevoegd. Elke emmer heeft een maximale en minimale waarde van wat erin mag. Wij hebben gekozen dat we 20 buckets maken in de eerste bucket kunnen er waardes zitten van 0 tot en met 4999 en in de laatste bucket kunnen de er waardes in zitten van 95000 tot 99999.

**private void** bucketSort(Player player){  
 **int** bucketToUse = (**int**)player.getHighScore() / ***MAX\_ARRAY\_SIZE***;  
 List<Player> bucket = **buckets**.get(bucketToUse);  
 bucket.add(player);  
 insertionSort(bucket);  
 **highScores**.clear();  
 **for** (**int** i = **buckets**.size() -1 ; i >= 0; i--) {  
 **highScores**.addAll(**buckets**.get(i));  
 }  
}

Wanneer er een highscore wordt toegevoegd wordt de volgende code uitgevoerd. We kijken eerst in welke bucket de highscore moet. Daarna voegen we de highscore toe aan die bucket en sorteren ook alleen die bucket. Dit is een optimalisatie, je hoeft doordat we onze buckets al van tevoren maken en bewaren niet steeds alle buckets te sorteren. Voor het sorteren van de bucket gebruiken we insertion-sort. Nadat het gesorteerd is gaan we door alle buckets en voegen elke highscore die in een bucket zitten toe aan de lijst.

## Grafiek

Hieronder vind je de grafiek die bij bucket-sort hoort. We zien hierin hoelang het duurt om het sorteren met bucket-sort. We beginnen met 100 highscores en verdubbelen de waarde steeds. De maximaal aantal highscores is 10000000. Maar wanneer het sorteren langer dan 15 seconden duurt gaan we niet verder.

# Priority queue

#TODO

# Conclusie

#TODO conclusie schrijven

#TODO grafiek erbij met alle waardes bij elkaar. Ik heb er nu 1 met 2 waardes maar daar moet priority qeue bij.