FUN3 – Algoritmes

1. Fisher–Yates (modern) shuffle algoritme
2. Ik heb voor deze algoritme gekozen omdat ik voor mijn Big Idea een Array van Strings wil husselen. In dit geval mogelijke antwoorden op een quizvraag waarbij 1 van de antwoorden correct is. Zodat de correcte antwoord telkens op een random positie komt te staan wil ik de array husselen.
3. **Pseudocode**

Initialiseer een lijst A

Initialiseer een integer N dat gelijk staat aan lengte van A

Initialiseer een random generator

**Loop**

(initialiseer een integer X dat gelijk is aan N - 1; loop loopt zolang X groter is dan 0; na elke loop gaat X met 1 omlaag)

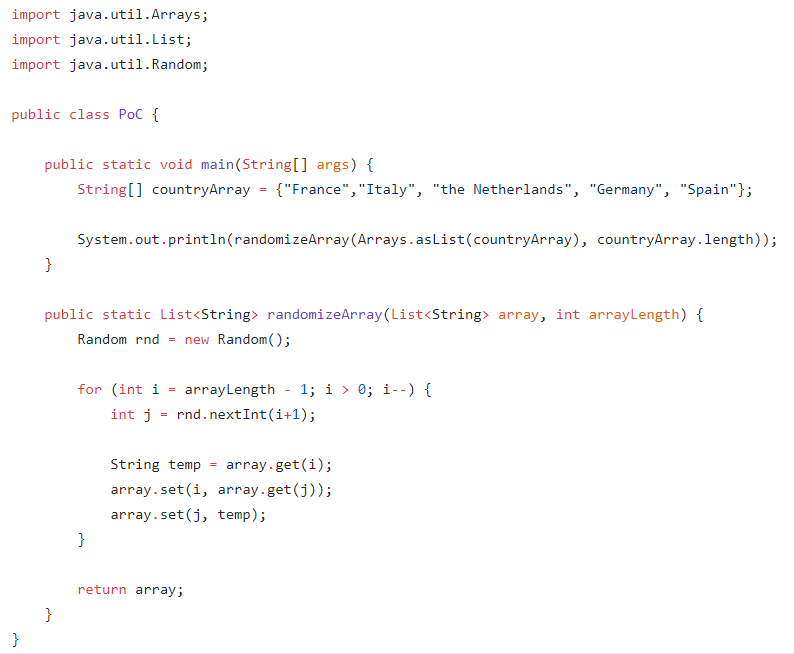
Initialiseer random integer Y tussen 0 - X+1

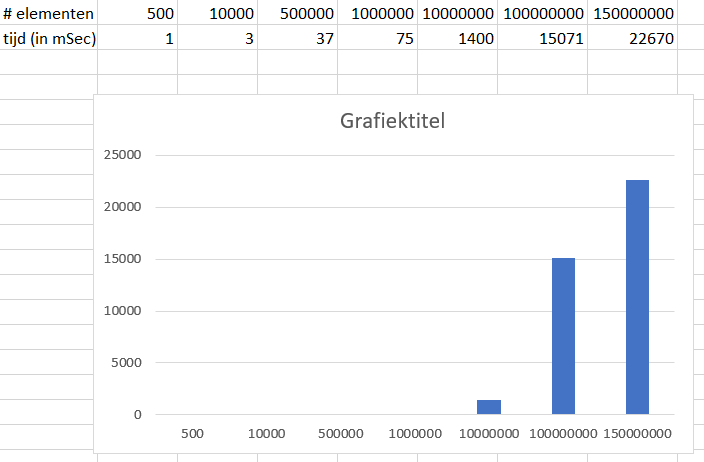
Initialiseer een integer die je gelijkstelt met element van A in positie Y

Initialiseer een integer W die je gelijkstelt met element van A in positie X

Wissel W met de element in A met positie Y

1. De time complexity van deze algoritme is O(n), wat inhoudt dat de complexiteit afhankelijk is van n, in dit geval van is dat de lengte van de gegeven array.
2. <https://github.com/eeffective/FUN3/blob/master/Algorithm/src/PoC.java>



1. 

**Reflectie**

Tijdens het maken van deze opdracht heb ik geleerd hoe je de performance/complexiteit van een algoritme kunt achterhalen door middel van “the Big Oh”. Er zijn verschillende algoritmen voor een probleem en ze hebben elk een eigen complexiteit.