project week 1

inleiding

in de eerste week hebben wij als groep de opdracht gekregen om een duurzaam project te ontwikkelen. hierover hebben wij eerst een presentatie gehad met daarin de informatie die nodig is voor de opdracht. hierover moeten wij als groep over brainstormen waaruit we een keuzen moeten maken wat wij willen doen.

# brainstorm

**dynamo regenpijp**

met behulp van regenwater en een dynamo willen wij energy opwekken.

voordelen:

* gratis energy naast het installeren van het systeem.

nadelen:

* de stroom opwek is waarschijnlijk zo minimaal waardoor het niet nuttig is.

**windturbine**

een windturbine die energy opwekt door wind

voordelen:

* de opwekking van energy is redelijk.

nadelen:

* het is al een bestaand product.

**smartshower**

met de smartshower willen wij mensen bewust maken over het verbruik van een douch. hieraan moet je denken aan verbruik van water en stroom.

voordelen:

* het verminderd verbruik van water
* het verminderd verbruik van energy

nadelen:

keuzen

uit de bovenste ideeën hebben we uiteindelijk gekozen voor de smart shower. hierin zagen we de meeste potentie. hiermee hopen we de gebruiker te overtuigen om korter en kouder te douchen. daarnaast zagen we hierin ook de meeste verduurzaming in. omdat je hier water en elektriciteit mee bespaard.

proces

07-02-2023

vandaag hebben we als groep vastgesteld wat we er wel en niet in willen hebben. dit hebben we aan de hand van een moscow methode gedaan. vanuit hier heeft iedereen zijn eigen stukje eruit gehaald om te gaan uitwerken.

vind kleinste nummer die x aantal keer voorkomt

voor deze opdracht heb ik gebruik gemaakt van een x aantal for loops. de eerste for loop zorgt ervoor dat hij blijft lopen zolang het array is. de tweede loop kijkt of het getal kleiner is dan het vorige getal (de eerste keer pakt hij gelijk het eerste getal). nadat hij alle getallen heeft vergeleken kijkt hij of het getal de gewenste aantal keer voorkomt. is dit niet het geval dan gaat hij opnieuw de loop in. om te voorkomen dat hij het zelfde getal weer als kleinste getal pakt, sla ik deze waarde op. in de if statement zeg ik dan als het getal kleiner is dan het huidige getal en groter is dan het al geteste getal dan moet je dit getal opslaan. door middel van deze manier zou hij de hele array aflopen tot hij het gewenste getal heeft gevonden of hij vind hem niet.

sorting methodes

voor het sorteren van een array kun je gebruik maken van een heleboel varianten. in mij test maak ik gebruik van quicksort, selection sort bubbelsorting.

bubbel sort

bij bubbel sort word er telkens gekeken of het getal groter is dan het getal ervoor. dit doet hij telkens opnieuw tot hij het aantal keer doorlopen gelijk is aan de array grote.

quick sort

quicksort pakt een getal en zet hem in een variabele pivot. dit getal is zelf in te stellen. met dit getal wordt de rest vergeleken. als het eerste getal dat groter is dan de pivot dan word deze opgeslagen. als dan een getal word gevonden dat kleiner is dan de pivot word deze geruild met de eerste getal die groter is dan de pivot. hierna begint hier weer met lezen vanaf het punt na de lage getal dat geruild is. wanneer alles wat kleiner is dan de pivot aan de rechter kant staat wordt de array opgesplitst in twee apparaten delen. dit blijft hij doen tot er maar 3 over zijn.

selection sort

selection sort loopt de array door en vergelijkt telkens of de een kleiner is dan het huidige getal. is dit het geval dan word de kleinere getal opgeslagen. wanneer de gehele getal is gelezen dan wordt het kleinste getal vooraan gezet en het eerste getal gaat naar de plek waar het laagste getal eerst stond. wanneer hij er voor de tweede keer in gaat start het met vergelijk vanaf de tweede getal omdat het eerste getal nu goed staat. heeft hij vanaf het tweede getal weer alles vergeleken dan komt het twee na kleinste getal op de tweede plek te staan en kijkt hij vanaf getal 3 enz.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| methode | bestand | Aantal x door loop | Tijd in ms | Op volgorde? | Aantal getallen |
| Bubbelsort 2 | In1\_1 | 15 | 0.000001 | nee | 5 |
|  | In1\_3 | 55 | 0.000001 | ja | 10 |
|  | In1\_5 | 705082704 | 10.321751 | ja | 100000 |
|  | In2\_4 | 499500 | 0.001534 | nee | 999 |
| Selectionsort 31 | In1\_1 | 14 | 1.000000 | nee | 5 |
|  | In1\_3 | 54 | 1.000000 | ja | 10 |
|  | In1\_5 | 705082703 | 3.810729 | ja | 100000 |
|  | In2\_4 | 499499 | 301.00000 | nee | 999 |
| Quicksort 32 | In1\_1 | 6 | 0.000001 | nee | 5 |
|  | In1\_3 | 45 | 0.000001 | ja | 10 |
|  | In1\_5 | 704982704 | 8.963972 | ja | 100000 |
|  | In2\_4 | 10510 | 0.000036 | nee | 999 |

Uit het tabel is te lezen dat quicksort over het algemeen het snelst is. Je kunt de pivot nog beter instellen waardoor de deling beter wordt gedaan.