Merlijn Folkerts, Elmar Hop, Lynn van Houttum, Lena Kemps

5vwo | wiskunde D

Sportmodules achter de computer

roosterproblemen oplossen

Inhoud

[Inleiding 2](#_Toc139028634)

[Probleemomschrijving 2](#_Toc139028635)

[Concept idee 1 3](#_Toc139028636)

[Uitwerking 1 3](#_Toc139028637)

[Concept idee 2 4](#_Toc139028638)

[Uitwerking 2 5](#_Toc139028639)

[Voorbereiding 5](#_Toc139028640)

[Uitvoering 7](#_Toc139028641)

[Het algoritme in code 9](#_Toc139028642)

[Stap 1: Zoek het volgende punt 9](#_Toc139028643)

[Stap 2: Check of het algoritme klaar is 10](#_Toc139028644)

[Stap 3: Voeg nieuwe punten toe 10](#_Toc139028645)

[Stap 4: Schakel het punt uit 11](#_Toc139028646)

[Opmerkingen 11](#_Toc139028647)

[Oude code 12](#_Toc139028648)

[Toepassing 14](#_Toc139028649)

[Invoer 15](#_Toc139028650)

[Conclusie 17](#_Toc139028651)

[Bronnenlijst 17](#_Toc139028652)

# Inleiding

In het vijfde jaar en de examenjaren van de middelbare school is het gymprogramma anders dan de jaren voorheen. In plaats van de docenten, kiezen de leerlingen wat zij gaan doen tijdens lichamelijke opvoeding.

Wij hebben gekozen voor het roosterprobleem van de gymmodules omdat het eenvoudiger is dan het schoolrooster. De lokalen, of in sommige gevallen, locaties, staan vast. De leraren zijn verbonden aan een module, in plaats van een klas. Ook lestijden zijn hier niet van belang, omdat de gymmodules altijd op een maandag- of donderdagmiddag plaatsvinden. We focussen ons enkel op de indeling van de modules, niet het regelen van de locatie of de docent.

Dat betekent niet dat het niet uitdagend kan zijn, omdat de gymmodules een uniek element hebben.

# Probleemomschrijving

Wat de gymmodules anders maakt dan een schoolrooster of toets week rooster, is dat de leerling uit de modules kiest.

Aan het begin van de gymmodules krijgen alle leerlingen een formulier. Voor 4 verschillende periodes worden minstens 5 modules aangeboden. Het is de taak van de leerling voor elke periode een top 3 van modules te maken.

Het is dan de bedoeling om zoveel mogelijk leerlingen in diens eerste keuze te plaatsen. Als een leerling niet in hun eerste optie geplaats kan worden, wordt die in hun tweede keuze geplaatst. Als dat ook niet mogelijk is, dan wordt de leerling in hun derde keuze ingedeeld. Geplaatst worden in de modules die niet gekozen zijn door de leerling is zeer ongewenst.

In een voorbeeldscenario zijn 5 mentorklassen, dus 150 leerlingen, de keuze gegeven tussen de 5 sporten: acrobatiek, basketbal, crossfit, darten en estafette, sporten a, b, c, d en e afgekort. Elk van de modules hebben een capaciteit van 40 leerlingen, dus plek voor 200 sporters. Met dit voorbeeldscenario maken en testen wij onze algoritme .

# Concept idee 1

We begonnen met het idee om aan elke keuze van een leerling een score te geven.

|  |  |
| --- | --- |
| Betekenis | Waarde van |
| 1e keuze van leerling | 0 |
| 2e keuze van leerling | 1 |
| 3e keuze van leerling | 2 |
| Niet gekozen door leerling | 4 |

Vanuit hier kunnen we een populariteitsscore berekenen en koppelen aan elke sport module. Hierna kijken we naar de sport module met de laagste score, dat is dus de meest populaire sport module. We kijken naar de leerlingen die de populairste sport als eerst keuze hebben en als dit aantal leerlingen groter is dan de maximale capaciteit, bepalen wie er geplaatst wordt door middel van hun tweede keuze. Dit doen we door te kijken welke leerlingen de sport module die de hoogste score heeft als tweede keuze hebben gekozen, en die leerlingen worden dan op hun tweede keuze geplaatst.   
Elke populariteitsscore wordt constant geüpdatet door het aftrekken van de score die is gekoppeld aan de keuze van de leerling die erin geplaatst wordt. Dit concept is gebruik in de eerste cyclus.

# Uitwerking 1

Wij hebben om naar bovenstaand algoritme toe te bouwen het eerst wat simpeler aangepakt; uiteindelijk kwamen we uit op 3 verschillende algoritmen waarvan de laatste het bovenstaande algoritme is.

**Algoritme 1:**

Dit algoritme maakt in het begin een lijst met alle leerlingen, en een lijst met alle sporten. Vervolgens vraagt het per sport aan de gebruiker om het max aantal leerlingen en koppelt deze informatie aan de bijbehorende sport. Dan, om de leerlingen in te delen stopt het eerst elke leerling bij hun eerste keuze tot deze vol is, de leerlingen die nog niet zijn ingedeeld worden dan ingedeeld bij hun tweede keuze, tenzij die ook vol is, dan gaan ze bij de derde, etc. Aan het einde worden de resultaten uitgeprint in de terminal.

**Algoritme 2:**

Dit algoritme is eigenlijk een toevoeging op het eerste algoritme, het doet in het begin alles hetzelfde, maar voordat het de resultaten uitprint optimaliseert het de indeling, dat doet hij zo:

Kies 2 leerlingen uit de lijst met leerlingen en draai hun indeling om, bereken de score van de nieuwe indeling. Als de nieuwe indeling een betere score heeft blijven de leerlingen bij hun nieuwe plaats. Dit wordt herhaald tot het niet beter kan.

**Algoritme 3:**

Dit algoritme is al uitgelegd. Het programmeren van dit algoritme ging moeizamer dan verwacht, en tegen de tijd dat we het afhadden hadden we een nieuw en verbeterd concept bedacht, dat hieronder wordt uitgelegd.

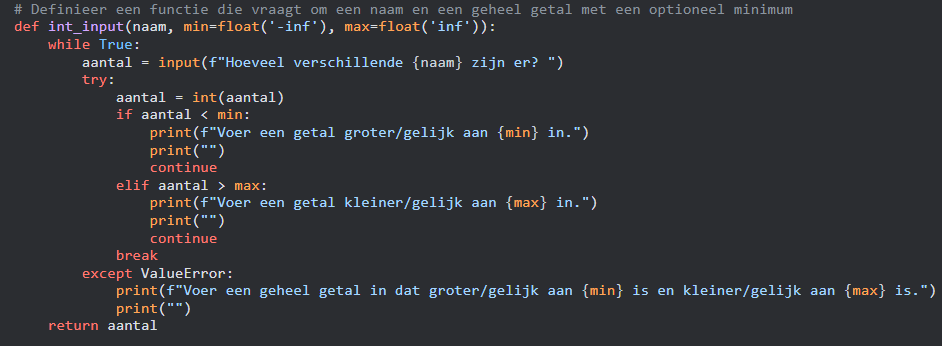
## Oude code

Keuzemaker  
Wij hebben een code in Python geschreven waarmee we de keuzes van leerlingen kunnen simuleren, zodat we de code voor het roostermaken kunnen uitproberen. Hierbij de uitleg van deze code.

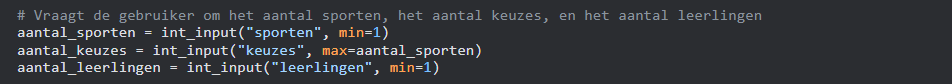
Line 1-2: Hiermee voeg je meer functies en mogelijkheden toe aan je standaard Python.

Inserting image...

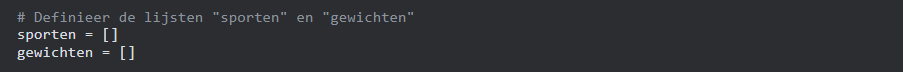
Line 5-23: Met deze code kun je een functie omschrijven die om een geheel getal vraagt met een minimum en maximum.



Line 25-28: Hiermee gebruik je de functie die eerder gedefinieerd om het aantal sporten, keuzes en leerlingen te vragen aan de gebruiker.



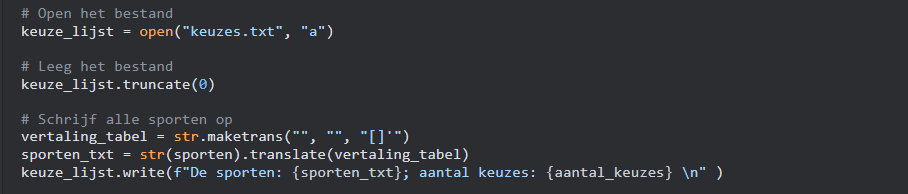
Line 30-32: Dit maakt 2 lijsten. Eén met de sporten die de gebruiker gaat invullen en één met de populariteit van de sporten bij de leerlingen.



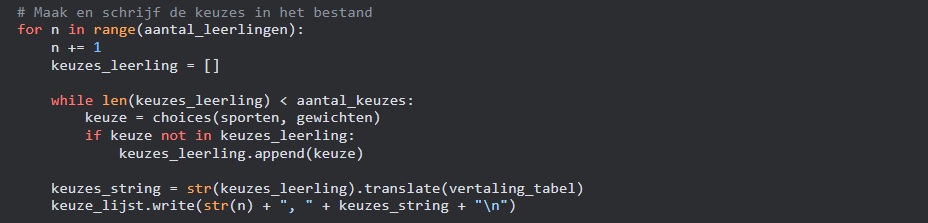
Line 34-56: Deze code vraagt de gebruiker per sport om de naam en de populariteit in te vullen.



Line 58-67: Deze code opent een tekstbestand dat “keuzes.txt” heet. Het leegt dit bestand en schrijft de namen van de sporten en het aantal keuzes per leerling op.   
Bijvoorbeeld. De sporten: a, b, c, d, e; aantal keuzes: 3



Line 69-80: Deze code maakt per leerling het eerder aangegeven aantal keuzes waarbij de populariteit van de sporten als gewicht voor deze keuzes gelden. Dit wordt vervolgens in het tekstbestand “keuzes.txt” geschreven.



<https://medium.com/swlh/tabu-search-in-python-3199c44d44f1>

# Concept idee 2

In ons geval maakt iedere leerling een top 3 uit 5 modules. Deze getallen kunnen echter verschillen. Hier is ons algoritme op aangepast.

Wanneer de leerling wordt ingedeeld in een module, wordt er een waarde aan de plaatsing gehangen. Hiervoor is een formule nodig om van een keuze naar een score te gaan. Hier kan elke formule voor gebruikt worden, mits is voor alle die gebruikt worden in het probleem. Hiernaast is het handig als strikt stijgend is voor gehele , zodat een slechtere keuze altijd een grotere score krijgt.

Voor ons probleem gebruiken wij de volgende formule:

Waarbij de score van de n-de keuze is. Hierbij begint vanaf 0, dus voor de 1e keuze geldt , voor de 2e keuze , enzovoorts. Voor de modules die een leerling niet gekozen heeft geldt  
, waarbij het totaal aantal modules is. Dit is omdat we van het slechtste geval uitgaan, namelijk dat dit de laatst gekozen module zou zijn geweest.

Hierbij is het belangrijk dat als het aantal keuzes van leerlingen is, altijd kleiner zal zijn dan .

Deze formule hebben we gekozen omdat deze stijgend is voor , en zodat de scores snel omhoog gaan als er een 3e keuze in de indeling zit, of zelfs een leerling die niet is ingedeeld in een module die die leerling heeft gekozen.

Als en , dan corresponderen de keuzes met de volgende waardes van :

|  |  |  |
| --- | --- | --- |
|  | Betekenis | Waarde van |
| 0 | 1e keuze van leerling | 0 |
| 1 | 2e keuze van leerling | 1 |
| 2 | 3e keuze van leerling | 4 |
| 4 | Niet gekozen door leerling | 16 |

Van alle leerlingen in alle modules kunnen de scores van de leerlingen opgeteld worden als indicator hoe goed het ingedeeld is.

Bijvoorbeeld als er een module bij zeven leerlingen als eerste keuze, bij twee leerlingen als tweede keuze en bij één leerling als derde keuze is ingedeeld, dan heeft deze module de volgende score:

De totale score van de indeling is gelijk aan de som van de scores van de modules. Het is de bedoeling om deze score zo laag mogelijk te krijgen. Deze scores worden gebruikt in de laatste modelleercycli.

# Uitwerking 2

## Voorbereiding

Om zeker te weten dat de beste indeling wordt gebruikt (nou ja, volgens de huidige scorering), wordt het Dijkstra-algoritme gebruikt. Dit algoritme kan op een gewogen graaf (een graaf waarbij aan elke lijn ook een gewicht kan worden toegekend, dit zou bijvoorbeeld een afstand kunnen zijn) het kortste pad vinden tussen 2 punten. Maar daarvoor moeten we eerst een graaf maken bij ons probleem.

Onze graaf is opgesteld volgens de volgende regels:

1. We beginnen met 1 punt. Dit representeert het moment waarop nog geen enkele leerling is ingedeeld.
2. Daarna maken we voor de eerste leerling voor elke module een punt aan. De volgende stappen worden uitgevoerd voor dit punt:
   1. Verbind het eerste punt met dit punt. Deze verbinding is gericht, zodat het eerste punt verbonden is met dit punt maar niet andersom.
   2. De lijn krijgt als gewicht de score die bij het totaal wordt opgeteld als de leerling in deze module zou worden ingedeeld.
3. Herhaal stap 2 voor elke leerling vanaf elk nieuw punt. Het is belangrijk dat er geen nieuw punt moet worden toegevoegd als de leerling niet meer in de module kan, omdat het via dit pad al vol zit.

Als voorbeeld nemen we even een versimpelde situatie waarbij we 3 leerlingen (A, B en C) hebben en 2 modules. De leerlingen hebben in dit voorbeeld allemaal dezelfde keuzes gemaakt, namelijk module 1 op de 1e plek, module 2 op de 2e plek. Dit is niet erg realistisch, maar het maakt het voorbeeld wat simpeler. Elke module kan in dit voorbeeld maximaal 2 leerlingen hebben.

We beginnen met 1 punt:

Afbeelding met schets, cirkel, clipart, ontwerp

Automatisch gegenereerde beschrijving

Daarna gaan we voor leerling A 2 nieuwe punten maken, 1 voor elke module:

Afbeelding met schermopname, cirkel, biljartbal

Automatisch gegenereerde beschrijving

Deze punten verbinden we met punt 0, maar slechts 1 kant op. We geven elk punt ook een gewicht gelijk aan de score van die keuze:

Afbeelding met cirkel, schermopname, zwart, kunst

Automatisch gegenereerde beschrijving

Omdat module 1 op de 1e keuze stond krijgt de lijn die naar module 1 gaat een gewicht van en de lijn naar module 2 een gewicht van .

Nu herhalen we dit voor leerling B. We maken voor elk punt die we net hebben toegevoegd weer een punt aan voor elke module:

Afbeelding met cirkel, schermopname, licht

Automatisch gegenereerde beschrijving

Nu weergeven punten 3 en 5 module 1, en punten 4 en 6 module 2.

Nu verbinden we de punten en geven we de lijnen een gewicht:

Afbeelding met cirkel, schermopname, kunst

Automatisch gegenereerde beschrijving

Omdat ook leerling B module 1 op de 1e plek heeft gezet en module 2 op de 2e, hebben de lijnen naar links (naar module 1) een gewicht van 0, en die naar rechts (naar module 2) een gewicht van 1.

Nu herhalen we dit voor leerling C. Hier is het ook nog belangrijk dat leerling C niet meer ingedeeld kan worden in module 1 als de andere leerlingen al in module 1 zitten, en hetzelfde geldt voor module 2. Deze punten worden niet toegevoegd.

Afbeelding met cirkel, schermopname

Automatisch gegenereerde beschrijving

Let hierbij op dat punt 7 voor module 2 staat en punt 12 voor module 1.

Dit is voor dit voorbeeld de complete graaf. Hieruit wordt wel duidelijk dat de graaf enorm groot wordt zodra er meer leerlingen en meer modules zijn. Vandaar dat voor dit voorbeeld een versimpeld probleem wordt weergegeven.

## Uitvoering

We gebruiken het Dijkstra-algoritme om een ‘perfecte’ indeling te maken (perfect voor de huidige scoreregels). Hierbij wordt gebruik gemaakt van de graaf die in de voorbereiding is gemaakt.

Voor een computer is het best veel werk om de hele graaf van tevoren te maken. Daarom worden punten pas toegevoegd zodra ze nodig zijn. Het is echter makkelijker om voor te stellen wat er gebeurt als alle punten er van tevoren al staan.

Het Dijkstra-algoritme wordt op de volgende manier toegepast:

We beginnen bij punt 0 (het punt waarbij nog geen enkele leerling is ingedeeld). Dit punt geven we een score van 0.

Afbeelding met cirkel, schermopname

Automatisch gegenereerde beschrijving

We geven nu elk punt dat direct verbonden is met het huidige punt een score die gelijk is aan de score van het huidige punt (in dit geval 0) plus de score van de lijn tussen het huidige punt en het andere punt. Dit is ook het moment dat het programma de punten toevoegt.

Afbeelding met cirkel, schermopname

Automatisch gegenereerde beschrijving

Nu gaan we naar het volgende punt. Dit is het punt met de laagste score dat nog niet verkend is.

Afbeelding met cirkel, schermopname, Graphics

Automatisch gegenereerde beschrijving

Dit proces herhalen we totdat we onderaan zijn aangekomen.

Afbeelding met cirkel, schermopname, Graphics

Automatisch gegenereerde beschrijving

Als er meerdere punten zijn met dezelfde score, maakt het niet uit welke wordt gekozen. Het is vaak het makkelijkst voor de computer om degene te pakken die we als eerste zijn tegengekomen. Dit verschilt echter afhankelijk van hoe het algoritme is geïmplementeerd.

Afbeelding met cirkel, schermopname, Graphics

Automatisch gegenereerde beschrijvingAfbeelding met cirkel, schermopname, Graphics, Kleurrijkheid

Automatisch gegenereerde beschrijvingAfbeelding met cirkel, schermopname, Graphics, Kleurrijkheid

Automatisch gegenereerde beschrijving

Nu zijn we onderaan aangekomen. Dat betekent dat we het pad hebben gevonden dat de laagste score geeft. Nu hoeven we alleen maar onze weg weer te volgen en dan hebben we de leerlingen ‘perfect’ ingedeeld.

De score kan niet lager vanwege het feit dat we weten dat dit het punt is met de laagste score. Als er een punt onderaan is met een lagere score, moet er wel een punt zijn die we hebben verkend met een score die onder onze score ligt. Dit is alleen niet zo, want ons punt had al de laagste score. Conclusie: er is geen pad met een lagere score. Het kan wel zijn dat er een ander pad is dat een gelijke score oplevert. Dit maakt niet echt uit, want die indeling is niet beter dan de indeling die wij hebben gevonden.

## Het algoritme in code

We hebben het algoritme geprogrammeerd in GDScript. Hiervoor is erg veel code gebruikt, wat betekent dat we hier niet alles kunnen laten zien. Alleen de meest belangrijke delen van de code staan hieronder beschreven. Het volledige programma, en oude code, kan gevonden worden op onze [GitHub](https://github.com/elmarhoppieland/Sportmodules).  
De code die is gebruikt is makkelijk aan te passen voor andere situaties waarbij een boom van punten wordt gebruikt voor het Dijkstra-algoritme. Hiervoor is de code opgesplitst in 2 delen: 1 deel voor onze specifieke situatie, en een ander deel is een plugin/addon, wat gemakkelijk gedeeld kan worden met andere Godot-projecten. Hier is de volgende code voor gebruikt:

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

De code herhaalt hierbij de volgende stappen tot hij klaar is:

1. Zoek het volgende punt met de laagste score
2. Kijk of het algoritme klaar is
3. Voeg nieuwe punten toe
4. Schakel het huidige punt uit

### Stap 1: Zoek het volgende punt

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Als *\_get\_next\_point()* is vervangen, wordt de waarde van die functie teruggegeven. Als dit niet zo is, pakt hij de beste child van de *root* van de boom.

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Deze functie gaat alle directe children langs, en wordt aangeroepen op al die children. Als het punt geen (directe of indirecte) children heeft die beter zijn (een lagere score hebben) dan het punt zelf, wordt het punt zelf teruggegeven.

Dit zorgt ervoor dat altijd het punt onder de *root* met de laagste score wordt teruggeven, en wordt gebruikt voor de volgende stappen.

### Stap 2: Check of het algoritme klaar is

Om te checken of het algoritme klaar is, is een functie klaargezet die alleen werkt als hij vervangen wordt. Dit is omdat het algoritme niet standaard weet wanneer hij klaar is, en dit verschilt per scenario.

In ons geval ziet de check er als volgt uit:



Deze functie is simpelweg één if-check, die kijkt of de index van de leerling van het (beste) punt de laatste is.

### Stap 3: Voeg nieuwe punten toe

Net als *\_check\_terminate()* hoort *\_create\_new\_points()* ook vervangen te worden. In ons geval ziet de code er zo uit:

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

De functie begint met een paar checks die ervoor zorgen dat we zeker weten dat het algoritme doet wat het hoort te doen.

Daarna gaan we alle modules langs, en voeren we de volgende acties uit:

1. Check of de module vol zit. Als dit zo is, hoeven we geen nieuw punt.
2. Maak een nieuw punt aan. Voor nu is dit punt ‘leeg’.
3. Geef het nieuwe punt een score afhankelijk van de score die de nieuwe leerling zou krijgen als diegene wordt ingedeeld in deze module.
4. Geef de leerling-index, module-index, en de grootte van de modules mee aan het nieuwe punt.
5. Voeg het punt toe als child van de *origin*.

### Stap 4: Schakel het punt uit

Dit is de simpelste stap. Elk punt heeft een variabele die kijkt of het punt aan of uit staat. Als een punt uit staat, doet dat punt net alsof hij een lagere score heeft dan alle andere punten. Deze stap houdt alleen maar in dat we deze variabele naar *true* zetten.

## Opmerkingen

* Het algoritme werkt niet als de graaf ook negatieve gewichten zou hebben, aangezien het dan zou kunnen dat een pad wat nog niet verkend is toch lager uit zou komen dan ons pad. Dit is een beperking van het Dijkstra-algoritme. Maar door een formule te kiezen voor waar voor geldt dat , zal de score nooit negatief worden.
* Dit is technisch gezien niet exact het Dijkstra-algoritme, aangezien we geen vast eindpunt hebben. Dit is op te lossen door alle punten onderaan de graaf (de punten waarvoor alle leerlingen zijn ingedeeld) te verbinden met een nieuw punt, met een lijn met een gewicht van 0. Dit zorgt er echter voor dat het algoritme langzamer wordt, aangezien het weer even kan duren voordat het laatste punt verkend wordt. Het is makkelijker om gewoon te zeggen dat het laatste punt bereikt is zodra we onderaan de graaf zijn.
* Het algoritme is niet perfect, en kan zeker nog wel verbeterd worden. Dit zou bijvoorbeeld kunnen met het A\* algoritme, wat een optimalisatie is van het Dijkstra-algoritme. Dit hebben we echter niet gedaan, aangezien dat nog veel meer tijd zou kosten. En zolang de capaciteiten een beetje ruim worden genomen, zal het algoritme iedereen vaak binnen 1 seconde kunnen indelen.
* Vaak zullen de leerlingen alfabetisch gesorteerd staan op de achternaam. Als de leerlingen niet eerst gehusseld worden, zal dit er echter ervoor zorgen dat de leerlingen met een achternaam die begint met bijvoorbeeld een ‘A’ vaker de 1e keuze krijgen dan een leerling met een ‘Z’. Dit is een probleem waar wij tegenaan liepen. Dit hebben we opgelost door eerst shuffle() uit te voeren op de leerlingen, en voordat de indeling wordt afgemaakt de leerlingen te sorteren op achternaam. Dit heeft als gevolg dat niemand een duidelijk voordeel heeft ten opzichte van iemand anders.

# Toepassing

Op de donderdagmiddag krijgen een aantal 5-vwo klassen sportmodules. Hierbij wordt voor alle modules een maximale capaciteit gehanteerd. De capaciteit die gehanteerd word is echter erg krap. Het algoritme kan hiervoor wel een indeling genereren, maar dit zal enorm lang duren. Daarom hebben wij voor elke module de maximale capaciteit iets verhoogd. Dit zorgt ervoor dat het algoritme binnen een paar minuten een indeling kan genereren.

In de eerste periode waren de sporten om uit te kiezen: schaatsen, fitness, Sixpack Challenge, volleybal en zaalvoetbal.

* Fitness heeft 23 als maximum capaciteit
* Sixpack Challenge heeft 23 als maximum capaciteit
* Volleybal heeft 31 als maximum capaciteit
* Zaalvoetbal heeft 31 als maximus capaciteit
* Schaatsen heeft 27 als maximum capaciteit

In de tweede periode waren de sporten om uit de kiezen: fitness, kickboksen, badminton, squash en body balance.

* Badminton heeft 39 als maximum capaciteit
* Fitness heeft 24 als maximum capaciteit
* Kickboksen heeft 27 als maximum capaciteit
* Squash heeft 24 als maximum capaciteit
* Body balance heeft 24 als maximum capaciteit

De derde periode de sporten: padel, skateboarden, veldvoetbal, tennis en veldhockey.

* Padel heeft 24 als maximum capaciteit
* Skateboarden heeft 24 als maximum capaciteit
* Veldvoetbal heeft 43 als maximum capaciteit
* Tennis heeft 24 als maximum capaciteit
* Veldhockey heeft 24 als maximum capaciteit

De vierde periode de sporten: mountainbike, softbal, crossfit, kanopolo en spelencircuit

* Mountainbike heeft 20 als maximum capaciteit
* Softbal heeft 35 als maximum capaciteit
* Crossfit heeft 24 maximum capaciteit
* Kanopolo heeft 24 als maximum capaciteit
* Spelencircuit heeft 35 als maximum capaciteit

De data van de keuzes is anoniem doorgegeven. Deze bijlage hebben wij van de gymdocenten gekregen om toe te passen. Deze bijlage staat op onze [GitHub](https://github.com/elmarhoppieland/Sportmodules) (of klik [hier](https://github.com/elmarhoppieland/Sportmodules/raw/main/invoerwerkblad%20do%207-8.xlsx) om het bestand te downloaden).

Met ons systeem hebben wij een indeling gemaakt. Deze is als bijlage toegevoegd. De score van deze indeling is 55 punten.

Hierbij is het belangrijk om op te merken dat wij misschien voor sommige modules de capaciteit hebben verhoogd tot een punt waarbij het misschien praktisch gezien niet werkt, omdat de school simpelweg echt niet zoveel leerlingen in een module kwijt kan. Het is dus lastig vergelijken met een indeling van de school.

# Invoer

Wij hebben een programma geschreven om te zorgen dat het makkelijk is om de gegevens in te voeren. Het programma kan dan het algoritme uitvoeren om een indeling te maken. Als het programma wordt opgestart ziet het er als volgt uit:

Afbeelding met tekst, schermopname, ontwerp

Automatisch gegenereerde beschrijving

Om het programma te gebruiken, moeten er eerst gegevens worden geïmporteerd. Dit kan door een Excel-sheet te kopiëren. De gekopieerde Excel-sheet moet er als volgt uitzien:

* Op de eerste rij komen de namen van de modules. Hierbij zijn een paar dingen om rekening mee te houden:

1. In het begin komen 3 lege cellen. Dit is omdat hier de klas, voornaam en achternaam van elke leerling komt.
2. Tussen periodes door moet 1 lege kolom zitten.
3. Aan het einde van de laatste periode moet 1 lege kolom zitten.

* Op de 2e rij komen de volgende cellen, in deze volgorde:

1. Klas
2. Voornaam
3. Achternaam
4. Voor elke periode komt ‘per 1’, ‘per 2’, … te staan. Dit kan door de cellen in die periode samen te voegen, of door de tekst in de 1e cel van de periode te zetten.

* Daarna komen alle leerlingen, met de klas, voornaam en achternaam in hun bijbehorende kolommen. De keuzes van de leerling staan bij de modules. Een 1e keuze krijgt een 1, een 2e keuze een 2, enzovoorts.

De cellen die gekopieerd moeten worden zijn die vanaf de lege cel linksboven, tot en met de lege cel rechtsonder.

Zodra de cellen zijn gekopieerd, kunnen de gegevens geïmporteerd worden met de knop ‘Van Klembord…’, of met ctrl-v. Het programma laat dan een tabel zien met de keuzes van alle leerlingen. Deze is hetzelfde als de tabel in Excel, maar dan zonder de opmaak.

Afbeelding met tekst, schermopname, menu

Automatisch gegenereerde beschrijving

Als de keuzes geïmporteerd worden, verschijnen er plekken om de capaciteit van de modules in te voeren. Het programma probeert een redelijke voorspelling te maken van de capaciteiten, maar dit kan nog aangepast worden.

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Met de capaciteiten van de modules ingevoerd kan de indeling gegenereerd worden met de knop ‘Genereer Indeling’. Het programma gaat dan even laden, en laat na afloop een tabel zien met de keuzes van alle leerlingen. Deze keuzes worden ook naar het klembord gekopieerd, zodat die makkelijk in een Excel-bestand geplakt kan worden.

Het is hierbij nog belangrijk om op te merken dat het algoritme enorm lang erover kan doen om de perfecte indeling te vinden als de capaciteiten erg beperkend zijn. Als het algoritme te lang duurt, kan dit opgelost worden door de capaciteiten ruimer te zetten. Als de capaciteiten echt zo strak moeten zijn, zal het algoritme er gewoon erg lang over doen. Dit kan soms een paar uur duren, of soms zelfs nog langer. Als de capaciteiten ruim genoeg zijn, kan er een indeling na slechts een paar seconden komen.

Als het programma er te lang over doet, kan het algoritme gestopt worden door het venster af te sluiten. Om het algoritme weer uit te voeren, kan het programma opnieuw worden opgestart.

# Conclusie

Met deze methode hebben wij ons gekozen roosterprobleem opgelost. De indelingen die onze uiteindelijke code maakt zijn optimaal. Door verschillende algoritmes toe te passen en te kijken wat de laagste scores genereert, hebben we gekeken welk algoritme het beste werkt voor ons roosterprobleem. Verder was de samenwerking in ons groepje goed en hebben we veel gebrainstormd om tot het eindproduct te komen. Iedereens ideeën zijn toegepast en serieus genomen en er is veel gedebatteerd over wat het beste resultaat zou geven. Onze code genereert niet alleen een optimale indeling met de laagste score, maar is ook gebruikersvriendelijk omdat het goed in combinatie met Excel te gebruiken is en omdat het programma makkelijk te begrijpen voor de normale gebruiker.

# Bronnenlijst

Spanning Tree. (2020, 15 Augustus). *How Dijkstra’s Algorithm Works* [Video]. YouTube. <https://www.youtube.com/watch?v=EFg3u_E6eHU>

# Bijlage: Onze indeling (anoniem)

Wij hebben de volgende indeling gegenereerd:

## Periode 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fitness | Sixpack Challenge | Volleybal | Zaalvoetbal | Schaatsen |
| 5V - Leerling 103 | 5V - Leerling 106 | 5V - Leerling 10 | 5V - Leerling 1 | 5V - Leerling 101 |
| 5V - Leerling 104 | 5V - Leerling 113 | 5V - Leerling 107 | 5V - Leerling 100 | 5V - Leerling 105 |
| 5V - Leerling 109 | 5V - Leerling 22 | 5V - Leerling 12 | 5V - Leerling 102 | 5V - Leerling 110 |
| 5V - Leerling 11 | 5V - Leerling 40 | 5V - Leerling 15 | 5V - Leerling 108 | 5V - Leerling 111 |
| 5V - Leerling 112 | 5V - Leerling 42 | 5V - Leerling 17 | 5V - Leerling 115 | 5V - Leerling 114 |
| 5V - Leerling 27 | 5V - Leerling 46 | 5V - Leerling 18 | 5V - Leerling 116 | 5V - Leerling 117 |
| 5V - Leerling 31 | 5V - Leerling 53 | 5V - Leerling 19 | 5V - Leerling 16 | 5V - Leerling 118 |
| 5V - Leerling 33 | 5V - Leerling 59 | 5V - Leerling 2 | 5V - Leerling 26 | 5V - Leerling 120 |
| 5V - Leerling 36 | 5V - Leerling 78 | 5V - Leerling 21 | 5V - Leerling 3 | 5V - Leerling 13 |
| 5V - Leerling 38 | 5V - Leerling 82 | 5V - Leerling 23 | 5V - Leerling 34 | 5V - Leerling 14 |
| 5V - Leerling 44 | 5V - Leerling 85 | 5V - Leerling 24 | 5V - Leerling 35 | 5V - Leerling 20 |
| 5V - Leerling 57 | 5V - Leerling 97 | 5V - Leerling 25 | 5V - Leerling 41 | 5V - Leerling 39 |
| 5V - Leerling 58 |  | 5V - Leerling 28 | 5V - Leerling 45 | 5V - Leerling 55 |
| 5V - Leerling 60 |  | 5V - Leerling 29 | 5V - Leerling 47 | 5V - Leerling 64 |
| 5V - Leerling 61 |  | 5V - Leerling 30 | 5V - Leerling 48 | 5V - Leerling 69 |
| 5V - Leerling 63 |  | 5V - Leerling 32 | 5V - Leerling 50 | 5V - Leerling 71 |
| 5V - Leerling 66 |  | 5V - Leerling 37 | 5V - Leerling 51 | 5V - Leerling 73 |
| 5V - Leerling 79 |  | 5V - Leerling 4 | 5V - Leerling 52 | 5V - Leerling 89 |
| 5V - Leerling 80 |  | 5V - Leerling 43 | 5V - Leerling 54 | 5V - Leerling 93 |
| 5V - Leerling 86 |  | 5V - Leerling 49 | 5V - Leerling 56 | 5V - Leerling 94 |
| 5V - Leerling 87 |  | 5V - Leerling 5 | 5V - Leerling 65 | 5V - Leerling 96 |
| 5V - Leerling 91 |  | 5V - Leerling 6 | 5V - Leerling 68 |  |
| 5V - Leerling 92 |  | 5V - Leerling 62 | 5V - Leerling 70 |  |
|  |  | 5V - Leerling 7 | 5V - Leerling 75 |  |
|  |  | 5V - Leerling 72 | 5V - Leerling 76 |  |
|  |  | 5V - Leerling 74 | 5V - Leerling 81 |  |
|  |  | 5V - Leerling 77 | 5V - Leerling 83 |  |
|  |  | 5V - Leerling 8 | 5V - Leerling 90 |  |
|  |  | 5V - Leerling 84 | 5V - Leerling 95 |  |
|  |  | 5V - Leerling 88 | 5V - Leerling 98 |  |
|  |  | 5V - Leerling 9 | 5V - Leerling 99 |  |

## Periode 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Badminton | Fitness | Body balance | Kickboksen | Squash |
| 5V - Leerling 1 | 5V - Leerling 106 | 5V - Leerling 104 | 5V - Leerling 103 | 5V - Leerling 101 |
| 5V - Leerling 10 | 5V - Leerling 11 | 5V - Leerling 112 | 5V - Leerling 105 | 5V - Leerling 102 |
| 5V - Leerling 100 | 5V - Leerling 18 | 5V - Leerling 16 | 5V - Leerling 107 | 5V - Leerling 108 |
| 5V - Leerling 115 | 5V - Leerling 19 | 5V - Leerling 21 | 5V - Leerling 109 | 5V - Leerling 110 |
| 5V - Leerling 118 | 5V - Leerling 2 | 5V - Leerling 22 | 5V - Leerling 111 | 5V - Leerling 116 |
| 5V - Leerling 15 | 5V - Leerling 20 | 5V - Leerling 4 | 5V - Leerling 113 | 5V - Leerling 117 |
| 5V - Leerling 17 | 5V - Leerling 28 | 5V - Leerling 45 | 5V - Leerling 114 | 5V - Leerling 12 |
| 5V - Leerling 23 | 5V - Leerling 33 | 5V - Leerling 5 | 5V - Leerling 120 | 5V - Leerling 14 |
| 5V - Leerling 24 | 5V - Leerling 40 | 5V - Leerling 50 | 5V - Leerling 13 | 5V - Leerling 3 |
| 5V - Leerling 29 | 5V - Leerling 42 | 5V - Leerling 6 | 5V - Leerling 26 | 5V - Leerling 30 |
| 5V - Leerling 31 | 5V - Leerling 44 | 5V - Leerling 69 | 5V - Leerling 27 | 5V - Leerling 56 |
| 5V - Leerling 32 | 5V - Leerling 46 | 5V - Leerling 85 | 5V - Leerling 36 | 5V - Leerling 65 |
| 5V - Leerling 34 | 5V - Leerling 47 | 5V - Leerling 87 | 5V - Leerling 37 | 5V - Leerling 68 |
| 5V - Leerling 35 | 5V - Leerling 55 | 5V - Leerling 9 | 5V - Leerling 38 | 5V - Leerling 71 |
| 5V - Leerling 41 | 5V - Leerling 59 | 5V - Leerling 91 | 5V - Leerling 39 | 5V - Leerling 72 |
| 5V - Leerling 48 | 5V - Leerling 60 | 5V - Leerling 94 | 5V - Leerling 43 | 5V - Leerling 73 |
| 5V - Leerling 49 | 5V - Leerling 63 | | 5V - Leerling 57 | 5V - Leerling 76 |
| 5V - Leerling 51 | 5V - Leerling 64 | | 5V - Leerling 61 | 5V - Leerling 8 |
| 5V - Leerling 52 | 5V - Leerling 7 | | 5V - Leerling 66 | 5V - Leerling 80 |
| 5V - Leerling 53 | 5V - Leerling 75 | | 5V - Leerling 79 | 5V - Leerling 83 |
| 5V - Leerling 54 | 5V - Leerling 77 | | 5V - Leerling 82 | 5V - Leerling 90 |
| 5V - Leerling 58 | 5V - Leerling 78 | | 5V - Leerling 89 | 5V - Leerling 92 |
| 5V - Leerling 62 | 5V - Leerling 84 | | 5V - Leerling 96 | 5V - Leerling 95 |
| 5V - Leerling 70 | 5V - Leerling 93 | | 5V - Leerling 97 | 5V - Leerling 98 |
| 5V - Leerling 74 |  |  |  |  |
| 5V - Leerling 81 |  |  |  |  |
| 5V - Leerling 86 |  |  |  |  |
| 5V - Leerling 88 |  |  |  |  |
| 5V - Leerling 99 |  |  |  |  |

## Periode 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Padel | Skateboarden | Veldvoetbal | Tennis | Veldhockey |
| 5V - Leerling 101 | 5V - Leerling 106 | 5V - Leerling 102 | 5V - Leerling 107 | 5V - Leerling 1 |
| 5V - Leerling 103 | 5V - Leerling 109 | 5V - Leerling 115 | 5V - Leerling 11 | 5V - Leerling 10 |
| 5V - Leerling 104 | 5V - Leerling 111 | 5V - Leerling 16 | 5V - Leerling 113 | 5V - Leerling 100 |
| 5V - Leerling 105 | 5V - Leerling 118 | 5V - Leerling 2 | 5V - Leerling 12 | 5V - Leerling 110 |
| 5V - Leerling 108 | 5V - Leerling 120 | 5V - Leerling 26 | 5V - Leerling 14 | 5V - Leerling 114 |
| 5V - Leerling 112 | 5V - Leerling 21 | 5V - Leerling 27 | 5V - Leerling 15 | 5V - Leerling 117 |
| 5V - Leerling 116 | 5V - Leerling 22 | 5V - Leerling 3 | 5V - Leerling 18 | 5V - Leerling 13 |
| 5V - Leerling 47 | 5V - Leerling 28 | 5V - Leerling 30 | 5V - Leerling 19 | 5V - Leerling 17 |
| 5V - Leerling 65 | 5V - Leerling 29 | 5V - Leerling 41 | 5V - Leerling 20 | 5V - Leerling 23 |
| 5V - Leerling 68 | 5V - Leerling 36 | 5V - Leerling 44 | 5V - Leerling 32 | 5V - Leerling 24 |
| 5V - Leerling 69 | 5V - Leerling 40 | 5V - Leerling 45 | 5V - Leerling 35 | 5V - Leerling 31 |
| 5V - Leerling 71 | 5V - Leerling 42 | 5V - Leerling 50 | 5V - Leerling 37 | 5V - Leerling 33 |
| 5V - Leerling 76 | 5V - Leerling 46 | 5V - Leerling 52 | 5V - Leerling 38 | 5V - Leerling 34 |
| 5V - Leerling 80 | 5V - Leerling 53 | 5V - Leerling 54 | 5V - Leerling 39 | 5V - Leerling 49 |
| 5V - Leerling 82 | 5V - Leerling 59 | 5V - Leerling 58 | 5V - Leerling 4 | 5V - Leerling 55 |
| 5V - Leerling 83 | 5V - Leerling 6 | 5V - Leerling 61 | 5V - Leerling 43 | 5V - Leerling 56 |
| 5V - Leerling 88 | 5V - Leerling 60 | 5V - Leerling 63 | 5V - Leerling 48 | 5V - Leerling 62 |
| 5V - Leerling 90 | 5V - Leerling 64 | 5V - Leerling 66 | 5V - Leerling 5 | 5V - Leerling 72 |
| 5V - Leerling 92 | 5V - Leerling 77 | 5V - Leerling 70 | 5V - Leerling 51 | 5V - Leerling 73 |
| 5V - Leerling 93 | 5V - Leerling 78 | 5V - Leerling 75 | 5V - Leerling 57 | 5V - Leerling 74 |
| 5V - Leerling 94 | 5V - Leerling 84 | 5V - Leerling 79 | 5V - Leerling 7 | 5V - Leerling 91 |
| 5V - Leerling 95 | 5V - Leerling 85 | 5V - Leerling 81 | 5V - Leerling 8 |  |
| 5V - Leerling 98 | 5V - Leerling 9 | 5V - Leerling 86 | 5V - Leerling 89 |  |
| 5V - Leerling 99 | 5V - Leerling 96 | 5V - Leerling 87 | 5V - Leerling 97 |  |

## Periode 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mountainbike | Softbal | Crossfit | Kanopolo | Spelencircuit |
| 5V - Leerling 103 | 5V - Leerling 1 | 5V - Leerling 102 | 5V - Leerling 101 | 5V - Leerling 10 |
| 5V - Leerling 118 | 5V - Leerling 104 | 5V - Leerling 108 | 5V - Leerling 105 | 5V - Leerling 100 |
| 5V - Leerling 18 | 5V - Leerling 107 | 5V - Leerling 114 | 5V - Leerling 106 | 5V - Leerling 11 |
| 5V - Leerling 20 | 5V - Leerling 112 | 5V - Leerling 13 | 5V - Leerling 109 | 5V - Leerling 113 |
| 5V - Leerling 29 | 5V - Leerling 116 | 5V - Leerling 16 | 5V - Leerling 110 | 5V - Leerling 115 |
| 5V - Leerling 47 | 5V - Leerling 12 | 5V - Leerling 2 | 5V - Leerling 111 | 5V - Leerling 17 |
| 5V - Leerling 53 | 5V - Leerling 14 | 5V - Leerling 21 | 5V - Leerling 117 | 5V - Leerling 19 |
| 5V - Leerling 63 | 5V - Leerling 23 | 5V - Leerling 22 | 5V - Leerling 120 | 5V - Leerling 32 |
| 5V - Leerling 65 | 5V - Leerling 24 | 5V - Leerling 36 | 5V - Leerling 15 | 5V - Leerling 33 |
| 5V - Leerling 71 | 5V - Leerling 26 | 5V - Leerling 37 | 5V - Leerling 28 | 5V - Leerling 39 |
| 5V - Leerling 73 | 5V - Leerling 27 | 5V - Leerling 38 | 5V - Leerling 31 | 5V - Leerling 40 |
| 5V - Leerling 76 | 5V - Leerling 3 | 5V - Leerling 4 | 5V - Leerling 46 | 5V - Leerling 41 |
| 5V - Leerling 80 | 5V - Leerling 30 | 5V - Leerling 43 | 5V - Leerling 49 | 5V - Leerling 42 |
| 5V - Leerling 83 | 5V - Leerling 34 | 5V - Leerling 44 | 5V - Leerling 64 | 5V - Leerling 5 |
| 5V - Leerling 90 | 5V - Leerling 35 | 5V - Leerling 45 | 5V - Leerling 77 | 5V - Leerling 54 |
| 5V - Leerling 92 | 5V - Leerling 48 | 5V - Leerling 50 | 5V - Leerling 78 | 5V - Leerling 55 |
|  | 5V - Leerling 51 | 5V - Leerling 57 | 5V - Leerling 82 | 5V - Leerling 59 |
|  | 5V - Leerling 52 | 5V - Leerling 6 | 5V - Leerling 84 | 5V - Leerling 60 |
|  | 5V - Leerling 56 | 5V - Leerling 7 | 5V - Leerling 93 | 5V - Leerling 62 |
|  | 5V - Leerling 58 | 5V - Leerling 79 | 5V - Leerling 94 | 5V - Leerling 69 |
|  | 5V - Leerling 61 | 5V - Leerling 87 | 5V - Leerling 96 | 5V - Leerling 81 |
|  | 5V - Leerling 66 | 5V - Leerling 9 |  | 5V - Leerling 85 |
|  | 5V - Leerling 68 | 5V - Leerling 95 |  | 5V - Leerling 86 |
|  | 5V - Leerling 70 | 5V - Leerling 98 |  | 5V - Leerling 89 |
|  | 5V - Leerling 72 |  |  | 5V - Leerling 91 |
|  | 5V - Leerling 74 |  |  | 5V - Leerling 97 |
|  | 5V - Leerling 75 |  |  | 5V - Leerling 99 |
|  | 5V - Leerling 8 |  |  |  |
|  | 5V - Leerling 88 |  |  |  |