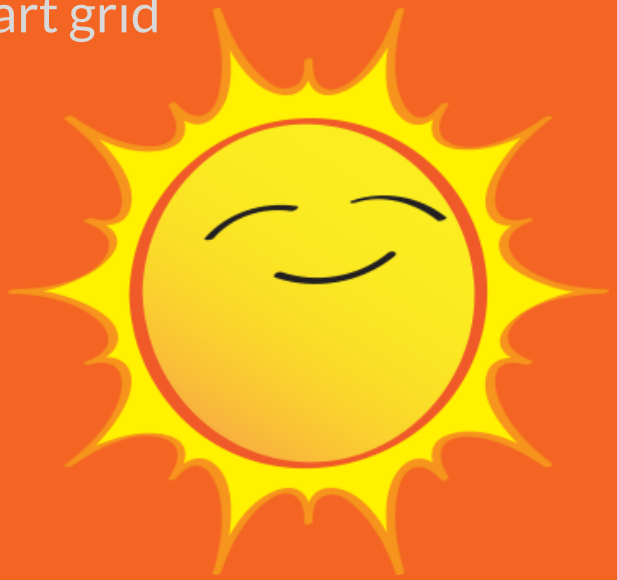
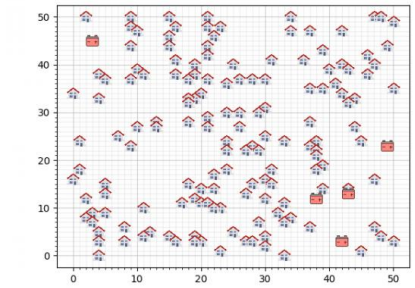
Sunny Storage

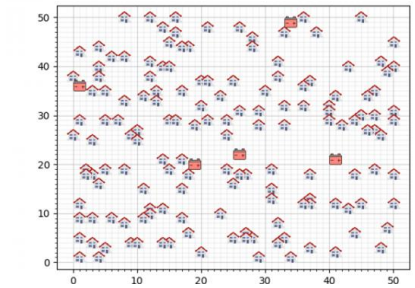
Smart grid

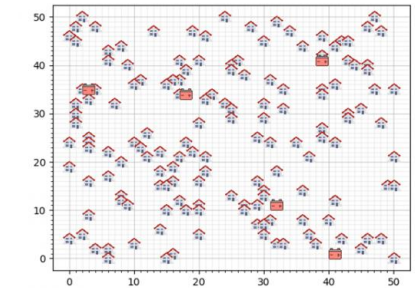


Bart, Feline en Jochem

Sunny reference
https://solarsystem.nasa.gov/system/basic_html_elements/11561_Sun.png







The case

- Drie wijken met huizen die maximale output genereren
- Deze moeten worden gekoppeld aan batterijen met een bepaalde capaciteit

1. Connect alle huizen aan een batterij
2. Leg kabel tussen de huizen en batterijen die connectie maken en bereken de kosten -> optimaliseer!
3. Verplaats batterijen/verzin nieuwe batterijen

Probleemstelling

- Verschillende algoritmes: welke geeft de ‘beste’ oplossing, i.e. kortste kabellengte/laagste kosten?

We weten $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ (Wiskunde A: nCr!)

Moeilijkheid ligt bij constraints...

Referentie formule: powerpoint van Daan (09-04-2019)

Probleemstelling

Moeilijkheid zit hem in contraints

- Batterijen mogen niet aan elkaar verbonden zijn
- Huizen mogen niet aan elkaar verbonden zijn
  - Tweede deel: kabel niet onder huis, anders boete 5000,-
- Huizen mogen niet aan meerdere batterijen verbonden zijn
- Het liefst kabels gescheiden houden, dit voorkomt kortsluiting
    - Geen harde eis

Probleemstelling vervolg

- Batterijen kunnen verplaatsen/zelf een verzinnen levert een ander probleem op: wat is de ideale configuratie voor de laagste prijs?

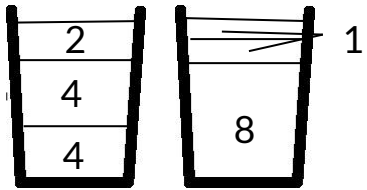
Constraint: je mag er zoveel plaatsen als je wilt, maar er zijn kosten aan verbonden

Update

- Deel 1: koppel alle huizen aan een batterij
- Versie van een 1D bin packing problem
  - Verdeel weights(outputs van huizen) over minimum aantal bins(batterijen)
- Min bins = som(weights)/capacity

$$\frac{7500}{\pm 1507} \approx 5$$

Weights: (4, 8, 1, 4, 2, 1)
Bin capacity: 10



- Benaderend algoritme

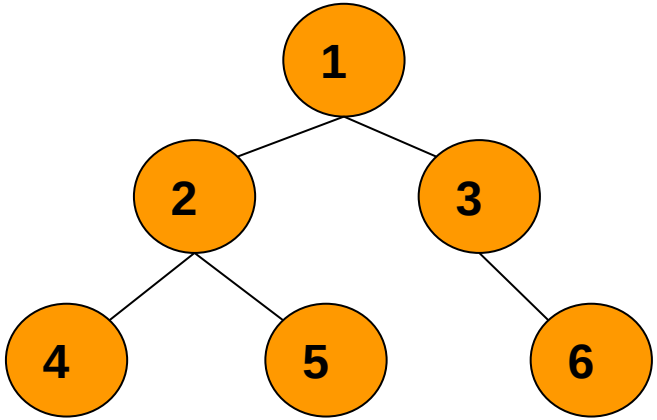
Benaderend algoritme

	Wijk 1	Wijk 2	Wijk 3
First fit			Fit 149/150
Decreasing first fit			Fit 149/150
Average fit			

- Als mogelijk nog een exact algoritme schrijven

Ideeën voor opdracht 2

Breadth first search (BFS) vs. Depth first search (DFS)



Ideeën voor opdracht 2

Algoritme A* :

- Kosten = kabellengte * prijs kabel per gridlijn
- A* zoekt naar de laagste kosten
  - Soms heb je meerdere paden met dezelfde lage kosten
    - Welke is dan het beste?
  - Het kost veel runningtime als je niet binnen de eerste tries je node vindt

Ideeën voor opdracht 2

Algoritme Breadth first search:

- Maakt een queue van alle children
- Je gaat de oplossing vinden
  - Runningtime wordt lang(er) bij grotere grid
    - Veel geheugen

To do!

- Met Angelo de datastructuur bespreken om te kunnen beginnen met algoritmes implementeren
- Nadenken over sorting/fitting methoden

References

https://www.slant.co/versus/11585/11586/~a-algorithm_vs_breadth-first-search

https://www.redblobgames.com/pathfinding/a-star/implementation.html#python

https://www.youtube.com/watch?v=ob4faIum4kQ

https://www.youtube.com/watch?v=KiCBXu4P-2Y&t=40s

https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2