Eindopdracht Programmeren 2.1 – De Stoelendans

Nu de corona maatregelen versoepeld zijn, mogen we weer naar concerten. In deze opdracht gaan we modelleren hoe mensen de stoelen in een concertzaal gaan bezetten. De concertzaal bestaat uit een vierkant rooster van vloer ('.'), lege stoelen ('L') en bezette stoelen ('#'). Bij aanvang van het concert (tijdstip $t = 0$) zijn alle stoelen leeg, zie het voorbeeld hieronder (deze opstelling staat in het bestand **test_input.txt**, je echte invoer **input.txt** is veel groter.)

```
............
.L.LL.LL.LL.
.LLLLLL.LL.
.L.L.L..L...
.LLLL.LL.LL.
.L.LL.LL.LL.
.L.LLLLL.LL.
...L.L......
.LLLLLLLLL.
.L.LLLLLL.L.
.L.LLLLL.LL.
............
```

$t = 0$

Mensen komen vervolgens de concertzaal binnen. Men is echter de versoepeling van de coronaregels nog niet helemaal gewend, en mensen proberen nog steeds drukte te vermijden. Ze geven de voorkeur om op plekken te zitten met weinig mensen om hun heen, en als het te druk wordt vertrekken ze weer. We modelleren de drukte door te kijken naar de acht plekken (horizontaal, vertikaal, en diagonaal) om een stoel heen.

- Als een stoel leeg is ('L') en niemand zit in de acht posities om de stoel heen, dan gaat er iemand zitten. (We modelleren de bewegingen van mensen, of de route die ze nemen *niet*, we doen alsof er iemand vanuit het niks op de stoel verschijnt.)
- Als een stoel bezet is ('#') en er zitten vier of meer mensen om hem heen dan vertrekt diegene die op deze stoel zit. (Ook hier modelleren we de beweging van mensen niet, we doen alsof diegene in het niks verdwijnt.)
- Niemand gaat op de vloer ('.') zitten en er worden geen stoelen bijgeplaatst.
- Anders blijft de bezetting van de stoel onveranderd.

Iedereen besluit op vaste tijdstippen of ze gaan zitten of vertrekken, onafhankelijk van de beslissingen van de mensen om hun heen.

Volgens deze regels is na één tijdstap (tijdstip $t = 1$) iedere stoel bezet:

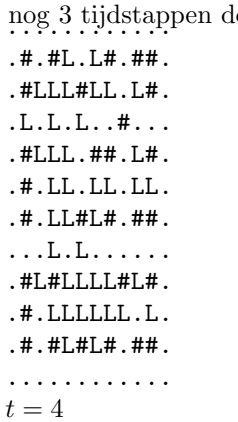
$t = 1$

Na een tijdstap later ($t = 2$) zijn alle stoelen met vier of meer stoelen er om heen weer leeg.

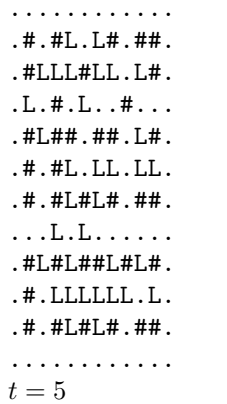
$t = 2$

Dit process gaat vervolgens nog 3 tijdstappen door.

$t = 3$



$t = 4$



$t = 5$

Na deze vijf stappen is de bezetting van de zaal stabiel, niemand gaat meer zitten, en niemand vertrekt. Er zitten nu 37 mensen.

De opdracht

Maak een programma dat het bovenstaande proces simuleert, met als begintoestand de lege zaal te vinden in het bestand **input.txt**.

- Houd voor iedere tijdstap bij hoeveel mensen er in de zaal zitten.
- Je programma gaat door totdat de zaalbezetting niet meer veranderd.
- Na afloop geeft je programma de volgende uitvoer:
  - Het aantal bezette stoelen.
  - Het print de zaalbezetting. (Bonuspunt: doe dit grafisch, met een plot, in plaats van met tekst uitvoer.)
  - Het toont een plot van het aantal mensen in de zaal tegen de tijd.

Tips

- Schrijf je programma zodat het met ieder mogelijk (.txt) invoer bestand kan werken. Je mag er wel van uit gaan dat de rand van de zaal uit vloer bestaat.
- Test je programma met het bestand **test_input.txt** en controleer of de bezetting na iedere stap hetzelfde is als hierboven beschreven.
- Er zijn meerdere manieren hoe je dit kan programmeren:
  1. De makkelijkste manier is om een 2D array (*nested list*) bij te houden van iedere positie in de zaal. Ieder element van deze lijst is een karakter ('.', 'L' of '#'). Vervolgens maak je voor iedere tijdstap een nieuwe lijst aan met de nieuwe bezetting. Dit is echter langzaam, je bent namelijk iedere tijdstap ook de vloer aan het nagaan, en deze veranderd niet.
  2. Een efficiëntere optie is om de stoelen te nummeren van 0 tot n, en voor iedere stoel bij te houden welke stoelnummers het als buren heeft. Vervolgens houd je een lijst van n bezettingen bij, en update je deze lijst iedere tijdstap.

  De tweede optie vergt wat meer programmeerwerk, maar is ongeveer twee keer zo snel.
- Splits je programma in kleinere onderdelen (Bijvoorbeeld: data inlezen, aantal stoelen om een positie tellen, een tijdstap doen), geïmplementeerd als functies die je afzonderlijk kan testen.

Beoordeling

Deze eindopdracht wordt individueel gemaakt, en beoordeeld op

- Correctheid code en uitvoer: 4 punten

Alleen als het aantal bezette stoelen en de geprinte zaalbezetting correct is worden de volgende punten nagekeken:

- Plot : 1 punt
- Programmeerstijl
  - Commentaar : 1 punt
  - Naamgeving variabelen : 1 punt
  - Structuur/Duidelijkheid : 1 punt
- Efficiëntie : 2 punten (zie Tips – De efficiënte oplossing is 1 punt.)
- Bonuspunt (grafische weergave zaalbezetting) (1 punt)

Cijfer is het totaal aantal punten.

Elkaar om hulp vragen is uiteraard toegestaan, een tweede paar ogen kan soms helpen die rare *bug* te vinden, maar iedereen schrijft zijn eigen code. Het overnamen van code van elkaar, of van het internet, is fraude en wordt gemeld bij de examencommissie.

Deadline

Lever je code in op BrightSpace op of voor zondag 14 november.