

## Casus 1 - Pakketdienst

Een pakketdienst heeft een grote vloot van bestelbusjes en diverse chauffeurs.

Dagelijks worden grote hoeveelheden pakketten van diverse afmetingen vanaf het distributiecentrum bezorgd bij allerlei geadresseerden.

Van een te vervoeren pakket is, vanaf het moment van ontvangst op het distributiecentrum tot en met het moment van bezorging bij de geadresseerde, de status bekend (distributiecentrum, onderweg, bezorgd, kwijt, beschadigd, etc.).

Voor een efficiënte dienstverlening zijn er verschillende problemen aan te wijzen die moeten worden opgelost, zoals bijvoorbeeld:

- Hoe kun je zo snel mogelijk de actuele status van een gegeven pakket opvragen?
- Wie zijn de top-10 geadresseerden in een afgelopen periode (dag, maand, ...)?
- Zijn alle gegevens in het juiste data format opgeslagen?
- Welke pakketten moeten er naar een bepaalde regio worden vervoerd?
- Gegeven een wegenkaart, wat is de snelste/kortste route van de huidige locatie naar een bepaald bezorgadres?
- Hoe maak je een efficiënte route voor een chauffeur langs meerdere gegeven adressen?
- Hoe laad je zoveel mogelijk pakketten van verschillende afmetingen in een busje (als je niet mag stapelen), gegeven de afmetingen van de laadvloer?
- Als er een grote hoeveelheid pakketten van verschillende afmetingen van het verzamelpunt naar een of meer adressen moet worden vervoerd, hoe verdeel je deze lading over een of meer busjes, zodat er met zo min mogelijk ritten kan worden volstaan?
- Hoeveel busjes/chauffeurs heb je per dag nodig?

Wellicht kun je zelf nog andere interessante vraagstukken bedenken die voor de pakketdienst van belang zijn.

Voor het oplossen van de vraagstukken van de pakketdienst heb je veel gegevens nodig, die je natuurlijk zo slim mogelijk wilt opslaan. Vervolgens zijn er verschillende algoritmen nodig om oplossingen te vinden voor de verschillende problemen.

Ter vereenvoudiging gaan we uit van een verspreidingsgebied waarin we alle adressen kunnen aangeven met een x-coördinaat en een y-coördinaat.

De wegenkaart heeft niet tussen elk tweetal adressen een rechtstreekse weg, er zijn slechts een beperkt aantal wegen. Verder kruisen wegen elkaar op diverse kruispunten (ook aan te geven met een x- en y-coördinaat).

Je mag zelf verschillende wegenkaarten definiëren, waarbij elke weg een lengte heeft en/of een tijdsduur om deze af te leggen.

Ter inspiratie vind je hieronder een voorbeeld met een distributiecentrum (D) en diverse bezorgadressen.



We hebben de volgende gegevens voor je beschikbaar:

- Het adres van het distributiecentrum (375,375)
- Een overzicht van klanten met hun adres (csv-bestand)
- Een overzicht van pakketten met afmetingen, datum van binnenkomst op distributiecentrum en bijbehorende geadresseerde

Ontwikkel voor de pakketdienst een *proof of concept* voor een applicatie die een aantal van hun vraagstukken/use cases oplost. Laat met behulp van de gegeven data, eventueel door jezelf aangevuld met gegeneerde data, zien dat de *proof of concept* naar behoren werkt.

In je *proof of concept* moeten in ieder geval elk van de volgende datastructuren en algoritmen zijn gebruikt:

- Een lineaire datastructuur en een binaire zoekboom, met in beide dezelfde gegevens opgeslagen. In beide datastructuren moet worden gezocht naar een specifiek element en de benodigde zoektijd moet worden gemeten en vergeleken bij verschillende (grote) aantallen opgeslagen gegevens.
- Een sorteeralgoritme.
- Een datastructuur die de gewenste functionaliteit efficiënt mogelijk maakt.
- Een graafstructuur met gegevens, waarop een of meer graafalgoritmen worden toegepast.
- Recursie en een backtracking algoritme.
- Eindige automaten en/of reguliere expressies.