

# Hoe een koe een haas vangt

November 3, 2018

## 1 Inleiding

Misschien ken je de uitdrukking "Je weet nooit hoe een koe een haas vangt". Volgens de Grote Van Dale betekent deze uitdrukking "je weet niet wat een onverwacht gelukje kan doen, zelfs wat onmogelijk lijkt, kan gebeuren"

In dit vak zullen we zien dat een koe inderdaad een haas kan vangen. Mits zij de beschikking heeft over kunstmatige intelligentie.

## 2 Opdracht

### 2.1 Tutorial

Als je ons framework, `libkmint`, wilt gebruiken, zorg er dan voor dat je de tutorial helemaal hebt gemaakt. Na afloop van de tutorial heb je namelijk een applicatie waarin een koe en een haas klaar staan op een graaf.

### 2.2 Van Dijkstra naar A\*

1. Laat de koe beginnen op het midden van de kaart (zie tutorial)
2. Zorg ervoor dat de haas op een willekeurig bepaald hoekpunt wordt geplaatst.
3. Zoek met Dijkstra de kortste route van de koe naar de haas en markeer alle knopen die je algoritme heeft bezocht. Hiervoor kun je de functie `tagged(true)` aanroepen op een knoop. Onderstaande code zal alle knopen op een oneven index in de graaf markeren. Knopen die gemarkeerd zijn worden in een andere kleur weergegeven:

```
for (std::size_t i = 0; i < graph.num_nodes(); ++i) {
    graph[i].tagged(i % 2 == 0);
}
```

Je kunt in een keer alle tags verwijderen met de functie `untag_all` van `map_graph`:

```
graph.untag_all();
```

4. Maak een schermafdruck van je applicatie. Zorg ervoor dat duidelijk is welke knopen gemarkeerd zijn en welke niet.
5. Zoek nu met A\* de kortste route (Zie ook hoofdstuk 5 van het boek). Markeer wederom alle knopen die het algoritme heeft bezocht. Denk bij het plannen van een route aan
  - (a) Een geschikte heuristiek
  - (b) Stel je zelf de vraag of deze heuristiek *admissible* is. Op het assessment moet je kunnen uitleggen waarom jouw heuristiek *admissible* is.
6. Maak wederom een schermafdruck
7. De koe moet rekening houden met het gewicht van de kanten. Als een kant een gewicht van twee heeft, moet de koe twee keer zo lang wachten voor zij zich via die kant kan verplaatsen.
8. Pas het A\*-algoritme aan opdat het rekening houdt met de gewichten.
9. Maak ook nu een schermafdruck.
10. Wat is het verschil in bezochte knopen in de schermafdrucken van stap vier, zes en negen?
11. Laat de koe het met A\* gevonden pad bewandelen tot zij de haas heeft gevangen
12. Laat de haas wanneer de koe haar heeft verspringen naar een andere knoop.
13. Programmeer de koe zo dat zij na het verspringen van de haas een nieuwe route bepaalt met A\* en deze gaat volgen.