

Eerste Deeltoets Algoritmiek

5 oktober 2010, 8.30 – 10.30, Educ. Gamma.

- Gelijke buur.** Kabouters zijn er in twee soorten, jongens en meisjes. Van een rij van n kabouters K_1 t/m K_n is gegeven: (1) het aantal, n , is *even*; (2) K_1 is een jongen en K_n ook.
 - Bewijs dat uit deze gegevens volgt, dat er ergens twee kabouters van gelijk type naast elkaar staan.
 - Generaliseer de gebruikte eigenschap tot een invariant.
 - Geef een iteratief algoritme dat een buurpaar van gelijk type vindt, door slechts $\lg n$ kabouters te bekijken.
- De Arme Monnik.** Een monnik gaat wandelen naar Santiago, wat L kilometer ver is; zijn maximale dagmars is M kilometer. Gelukkig zijn er onderweg herbergen H_1 t/m H_n , waarbij H_i ligt op s_i kilometer van het begin, en een overnachting kost daar c_i . Voor het gemak noemen we het begin $0 = s_0$, en het eind $L = s_{n+1}$, en zijn de herbergen alvast opgenoemd op volgorde van afstand: $s_i < s_{i+1}$. De herbergen liggen niet verder van elkaar dan de maximale dagafstand: $s_{i+1} \leq s_i + M$. De monnik wil een zo goedkoop mogelijke selectie van herbergen.
 - Beschrijf het bepalen van de goedkoopste reis als optimaliseringsprobleem (zoekruimte, toelaatbaarheid, doelfunctie).
 - Formuleer en bewijs een Optimal SubStructure Property.
 - Geef een Dynamisch Programmeer-algoritme dat de goedkoopste reis uitstippelt.
- Met de Bus.** Een busbedrijf beschikt over n bussen, waarbij bus i , c_i passagiers kan vervoeren (capaciteit). Er komt een aanvraag voor het vervoer van een groep van K personen. Omdat met name de chauffeurskosten hoog zijn, wil men de groep met *zo weinig mogelijk* bussen vervoeren.
 - Beschrijf zoekruimte, toelaatbaarheid en doelfunctie voor dit probleem.
 - Geef een efficient algoritme dat dit oplost. Formuleer en bewijs een eventueel gebruikte GCP, OSS of OSP.
- Geamortiseerde Boomkleuring.** Op een boom (met n knopen) moet een bewerking worden toegepast, die bestaat uit het kleuren van de knoop en het volledige pad naar de wortel. Als je bij het kleuren een reeds gekleurde knoop tegenkomt kun je stoppen, immers, alle knopen daarboven zijn ook al gekleurd.

```
methode KleurPad(knoop i):
    while (color[i] == white)
        { color[i] = black;
          if (p[i] != nil) {i = p[i]} }
```

Als kosten van deze operatie rekenen we het aantal malen dat de test in de while-statement wordt uitgevoerd.

- Laat met voorbeelden zien dat de kosten van `KleurPad` kunnen variëren van 1 tot $d + 1$ (d is het maximale aantal knopen op een pad in de boom).
- Definieer Φ als het aantal witte knopen in de boom en bewijs, dat elke `KleurPad` geamortiseerde prijs 1 heeft met betrekking tot Φ .
- Bewijs dat een reeks van m `KleurPad` operaties ten hoogste $n + m$ kost.