De eerste programmeeropgave — Schuiven
Algoritmiek voorjaar 2008, Universiteit Leiden

DEEL 1

Toestand-actie-ruimte

Deze programmeeropdracht gaat over (gegeneraliseerde) eenpersoons schuifpuzzels.
Bekend is de zogenaamde 15-puzzel: in een 4 bij 4 rooster staan de getallen 1 tot en met 15 in een willekeurige volgorde. Verder is er één lege plek. Je mag een getal verschuiven naar de lege plek, mits dat getal er horizontaal of verticaal direct aan grenst. Het doel is om (zo snel mogelijk) de volgende stand te bereiken:

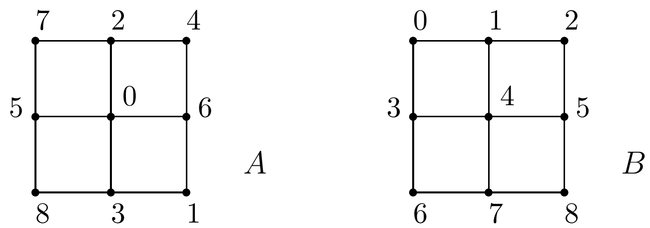
	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Merk op dat er $16! = 20922789888000$ verschillende toestanden zijn. Hiervan blijkt altijd precies de helft bereikbaar vanuit een gegeven begintoestand.

Geheel analoog kunnen we de 3-puzzel, de 8-puzzel, de 24-puzzel, … definiëren. We kunnen het probleem generaliseren tot rechthoekige schuifpuzzels, of zelfs tot grafen.

Gegeven een samenhangende ongerichte graaf met n knopen. In alle knopen zit een uniek getal (0 tot en met $n-1$) opgeslagen, in een of andere volgorde. Dat getal noemen we het *label* van de knoop. Het label 0 is speciaal: je mag een label naar een buurknoop schuiven als die buurknoop label 0 heeft. De vraag is dan of je, door herhaald te schuiven, een gegeven andere verdeling van de getallen over de knopen kunt bereiken, dat wil zeggen een andere *labelling* van de knopen. En zo ja, wat is dan de snelste manier? Voor een gegeven graaf met n knopen zijn er $n!$ labellingen mogelijk, corresponderend met alle mogelijke verdelingen van de labels 0 tot en met $n-1$ over de knopen.

Voorbeeld. Gegeven de volgende twee labellingen van de graaf $\mathcal{G}$ die een 3 bij 3 rooster modelleert:

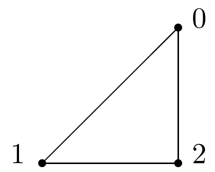


Kun je vanuit toestand (= labelling van $\mathcal{G}$) A toestand B bereiken door volgens de spelregels de getallen rond te schuiven? Merk op dat deze graaf met labellingen correspondeert met standen van de 8-puzzel.

Bij deze programmeeropdracht moet een kort **verslag** gemaakt worden, waarin de volgende vragen/opdrachten beantwoord/uitgewerkt moeten worden. Het verslag is bij voorkeur getypt (bijv. in LaTeX), maar mag ook handgeschreven zijn; als het maar duidelijk is, zowel wat betreft inhoud als leesbaarheid.

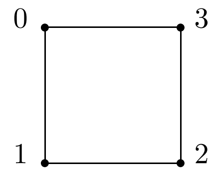
1. Teken voor onderstaande drie grafen met aangegeven labelling van de knopen (het deel van) de toestand-actie-ruimte vanuit die beginsituatie. Naast elke graaf staat de betreffende toestand in een verkorte notatie. Hint bij het overzichtelijk tekenen: teken de toestanden nivo voor nivo en nummer de verschillende toestanden. Dezelfde toestand hoef je maar één keer uit te werken. Vraag: zijn alle mogelijke toestanden bereikbaar vanuit de gegeven begintoestand?

a.



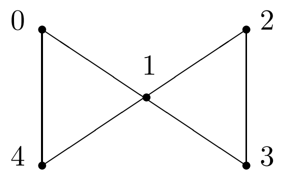
  0
1 2

b.



0 3
1 2

c.



0 2
  1
4 3

2. Geef de serie van achtereenvolgende verschuivingen die je moet doen om zo snel mogelijk vanuit stand I in stand II (zie hieronder) te komen. De onderliggende graaf is die uit **b**.

I: 0 1
    2 3

II: 3 2
     0 1

3. <u>Beredeneer</u> hoe je zo snel mogelijk vanuit stand III in stand IV (zie hieronder) kan komen. Gebruik hierbij alleen het deel van de toestand-actie-ruimte dat je in **1.** hebt getekend. De onderliggende graaf is weer die uit **b**.

III: 0 2
      1 3

IV: 3 0
     2 1

4. De volledige toestand-actie-ruimte die alle labellingen van de graaf uit **b.** bevat valt uiteen in twee samenhangscomponenten. Beredeneer hoeveel samenhangscomponenten de volledige toestand-actie-ruimte behorend bij de graaf uit **c.** heeft.

Programma

Er moet een (brute force) programma worden geschreven dat voor een samenhangende graaf met bijbehorende beginlabelling een lijst oplevert met alle toestanden (= labellingen van de graaf) die vanuit die beginsituatie te bereiken zijn. Verder moet berekend worden hoeveel verschuivingen de kortste "route" tussen twee labellingen gebruikt. De gebruiker moet het spel voor rechthoekige schuifpuzzels kunnen spelen.