Het algoritme van Kruskal

Op blz. 87-89 van het boek wordt een gretig algoritme beschreven voor het bepalen van een minimale opspannende boom. Dit algoritme wordt het algoritme van Prim genoemd. Er is een nog gretiger algoritme voor het zelfde probleem: het algoritme van Kruskal.

Net als bij het algoritme van Prim gaan we uit van een samenhangende, ongerichte graaf met gewichten (afstanden, kosten) op de takken. Ook beginnen we met een oorspronkelijk lege boom. Laat N het aantal knopen in de graaf zijn. Bij het algoritme van Kruskal kiezen we iedere stap de tak met het laagste gewicht die nog over is; als die tak geen kring vormt met eerder toegevoegde takken, voegen we deze toe aan de 'boom'. Vormt de tak wel een kring, dan gooien we hem weg. Als er bij een bepaalde stap meerdere takken met minimaal gewicht zijn, mogen we willekeurig een van die takken kiezen.

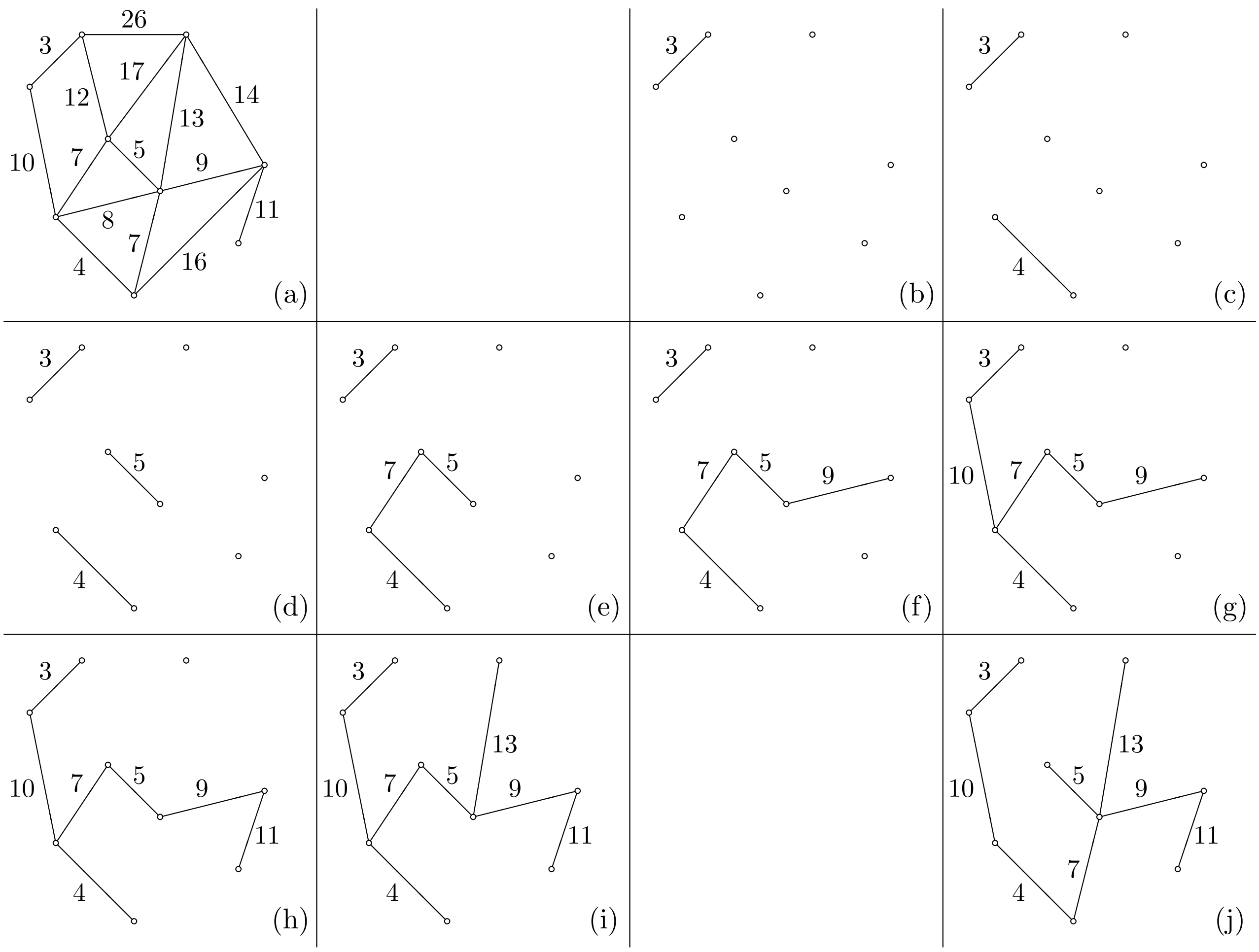
We stoppen als alle N knopen met elkaar verbonden zijn (d.w.z.: als er $N-1$ takken zijn toegevoegd). Eventueel kun je op dat moment nog doorgaan met proberen van volgende takken, maar die zullen altijd een kring vormen, dus dat heeft weinig zin.

Het resultaat van het algoritme is netjes een minimale opspannende boom. Het maakt hiervoor niet uit welke tak je kiest, in een stap waar je uit meerdere takken met minimaal gewicht kan kiezen. Verschillende keuzen kunnen uiteindelijk toch tot dezelfde boom leiden, maar ze kunnen ook verschillende bomen opleveren. In het laatste geval zullen al die bomen minimale opspannende bomen zijn.

Merk op dat de toegevoegde takken in tussenresultaten niet met elkaar verbonden hoeven te zijn. Bij het algoritme van Prim moeten de toegevoegde takken wel steeds met elkaar verbonden zijn.

In Figuur 1 werken we het algoritme van Kruskal uit voor een voorbeeld. Op twee gewichten na is de graaf in dit voorbeeld gelijk aan de graaf in het voorbeeld op blz 88-89 van het boek. In de vierde stap van het algoritme kunnen we kiezen tussen twee takken met gewicht 7. Bij de keuze die we in Figuur 1(e) maken, krijgen we de minimale opspannende boom in Figuur 1(i). Als we de andere tak met gewicht 7 hadden gekozen, zouden we de boom in Figuur 1(j) gekregen hebben.

Als we de tak met gewicht 5 in de graaf weer (net als in het boek) gewicht 15 geven, kunnen we nog steeds op een bepaald moment kiezen tussen twee takken met gewicht 7. Ga na dat we in dat geval bij beide keuzen dezelfde minimale opspannende boom krijgen.



Figuur 1: Werking van het algoritme van Kruskal. (a) Een samenhangende, ongerichte graaf met gewichten op de takken. (b)-(i) Er worden stap voor stap acht takken met minimaal gewicht toegevoegd aan de boom. Het resultaat in (i) is een minimale opspannende boom met totaal gewicht 62. (j) De minimale opspannende boom die we gekregen hadden, wanneer we in de vierde stap de andere tak met gewicht 7 hadden gekozen.