

Tweede deelloets Algoritmiiek (INFOAL) 11-11-2010

Motiveer je antwoorden.

Opgave 1. Paden in een gerichte graaf.

Als je in een gerichte graaf, met zeker startpunt s en eindpunt t , wilt bepalen of er een pad van s naar t bestaat, kun je kiezen tussen bijvoorbeeld BFS, DFS en Dijkstra's Kortste Pad algoritme.

- Wat is van elk van deze alternatieven de complexiteit (rekentijd)?
- Is het ook mogelijk om hiervoor een Minimum Spanning Tree algoritme te gebruiken (Kruskal of Prim-Dijkstra)?
- Als je een pad zoekt voor het FordFulkerson Max Flow algoritme, wordt vaak BFS aanbevolen. Waarom?

We kijken nu naar de situatie, waarin een deel van de kanten is gemarkeerd als ongewenst: de **rode** kanten.

- Kun je een van de algoritmen aanpassen om te bepalen of er een pad van s naar t *zonder rode kanten* bestaat?
- Kun je een van de algoritmen gebruiken om te bepalen of er een pad van s naar t bestaat met precies één rode kant?
- Kan dit in $O(n + m)$ tijd (n, m : aantallen knopen en kanten)?
- Kun je een van de algoritmen gebruiken om een pad te bepalen met een minimaal aantal rode kanten? Wat wordt de complexiteit?

Opgave 2. Randomiserende algoritmen.

Bij randomiserende algoritmen wordt vaak een onderscheid gemaakt tussen *Monte Carlo* en *Las Vegas* algoritmen. In deze opgave is $S[1 \dots n]$ een integer rij van *Type A* of *Type B*. In een *Type A* rij zijn *alle* getallen even; in een *Type B* rij is minstens de helft oneven.

- Wat is de definitie van een Monte Carlo algoritme?
- Wat is het beste *deterministische* algoritme om te bepalen of S van type A of B is?
- Wat is het beste Las Vegas algoritme om te bepalen of S van type A of B is?
- Je opdrachtgever wil een snelle test die het type bepaalt, met een betrouwbaarheid van 99%. Geef een Monte Carlo algoritme dat hieraan voldoet.
- Wat is van het algoritme uit d) de rekentijd? Is dat verwachte tijd of worst case? Beredeneer dat je algoritme aan de wensen van je klant voldoet.
- Is je algoritme een *A-test* of een *B-test*?
- Is er een verbetering mogelijk als vantevoren gezorgd wordt dat S gesorteerd is?

Opgave 3. Complexiteit.

In een project loop je aan tegen een optimaliseringsprobleem waarvoor jullie geen snel algoritme kunnen vinden. Een van de teamleden suggereert, aangezien jullie toch echt heel slim zijn, dat er dan misschien geen snel algoritme bestaat.

- a) Wat kun je doen om vast te stellen dat er geen snel algoritme bestaat? Geef enkele details over hoe dat precies werkt.
- b) Als inderdaad wordt vastgesteld dat het probleem niet snel op te lossen is, hoe kun je dan in je project met dit probleem omgaan?