Tentamen Gevorderde Algoritmen en Datastructuren

dinsdag 1 februari 2011, 18.30 - 21.30 uur

Je heoft slechts 4 van de 5 opgaven te maken: de opgave met het minste aantal punten telt niet mee. Elke opgave levert maximaal 25 punten op. Het tentamencijfer T is (p/10), waarbij p de som van de vier hoogste aantallen punten per opgave is.

Met de zinsnede 'geef een algoritme' in een opgave wordt bedoeld:

> **beschrijf een algoritme in pseudocode, licht de werking ervan toe, beargumenteer de correctheid.**

1. Geef het algoritme quicksort om een lijst van gehele getallen te sorteren. Beargumenteer dat, in het slechtste geval, de tijdscomplexiteit van het algoritme *niet* in $\mathcal{O}(n \log(n))$ is.
   Aanwijzing: het idee is om, aan de hand van een getal a in de lijst, de lijst te verdelen in drie deellijsten: groter dan a, gelijk aan a, kleiner dan a.

2. Geef het algoritme DFS(G, v) voor *depth-first search* vanuit knoop (*node*) v in een ongerichte samenhangende graaf G. DFS kent aan elke kant (*edge*) een label *discovery* of *back* toe: de *discovery edges* vormen een opspannende boom van G. Laat zien dat, bij een geschikt gekozen representatie van G, de complexiteit van het algoritme $\mathcal{O}(n + m)$ is, met m het aantal kanten en n het aantal knopen van G.
   Aanwijzing: gebruik recursie, en label ook de knopen op een geschikte wijze.

3. Gegeven zijn n positieve gehele getallen $c_1, \cdots, c_n$ en een positief geheel getal K. Geef een algoritme dat vaststelt of er een deelverzameling S van $\{1, \cdots, n\}$ is met de eigenschap

   $$\sum_{i \in S} c_i = K.$$

   Beargumenteer dat de tijdscomplexiteit in $\mathcal{O}(nK)$ is.
   Aanwijzing: gebruik dynamisch programmeren.

4. Gegeven is een string S van lengte n over een alfabet met d karakters. Beschrijf de suffix trie T van S met geheugenbeslag $\mathcal{O}(n)$, waarmee we in $\mathcal{O}(dk)$ tijd kunnen nagaan of een string P met lengte k als deelstring in S voorkomt. (Het algoritme voor zoeken in T hoeft niet in pseudocode gegeven te worden, een globale beschrijving van de werking volstaat. Vergeet de argumentatie mbt. de tijdscomplexiteit niet.)

5. Geef definities van de complexiteitsklassen P (polynomiaal) en NP (nondeterministisch polynomiaal). Laat zien dat P bevat is in NP. Beargumenteer waarom veel deskundigen denken dat NP niet bevat is in P.