Tentamen Algoritmen en Datastructuren in C

vrijdag 8 april 2011, 9 - 12 uur

1. (25 punt) Deze opgave gaat over de implementatie in C van een stack door middel van een gelinkte lijst.

   (a) Geef een `typedef` voor een gelinkte lijst met per knoop twee gehele getallen. Geef hierbij functies `isEmpty`, `push`, en `pop` om achtereenvolgens te testen of de lijst leeg is, een paar toe te voegen, en het laatst toegevoegde paar uit de lijst te halen en terug te geven (en het niet meer gebruikte geheugen vrij te geven).

   (b) Geef vervolgens een functie `bouw`, die voor een gegeven natuurlijk getal n de stack opbouwt van de paren $(1, 0)$ tot en met $(n, 0)$ in deze volgorde, en een functie `breekaf` die een gegeven stack van paren afbreekt en daarbij het eerste element van elk paar afdrukt. Gebruik hierbij de eerder gedefinieerde functies.

2. (25 punt)

   (a) Leg uit wat een *heap* is en hoe deze door een array kan worden gerepresenteerd.

   (b) Gegeven is de volgende C-code voor de implementatie van een heap die getallen bevat.

```
int maxSize = 2;
int front = 1;
int *heapArray;
heapArray = malloc(maxSize*sizeof(int));
assert(heapArray != NULL);
```

   Definieer nu in C de functies `enqueue`, `heapFull` en `upheap`. `enqueue(n)` voegt het getal n toe aan de heap, en maakt daarbij gebruik van `heapFull` (om indien nodig de omvang van de heap te verdubbelen) en `upheap` (om het heap-zijn weer te herstellen na de toevoeging van n).

   (c) Wat is de tijdscomplexiteit van `upheap`?

Z.O.Z.

3. (25 punt) Deze opgave gaat over ongerichte grafen.

   (a) Wanneer is een graaf *samenhangend*?

   (b) Wanneer is een graaf een *boom*? Wat is een *opspannende boom* van een graaf?

   (c) Beschrijf het algoritme Depth-First Search (DFS) in pseudocode. Dit algoritme doorzoekt een samenhangende graaf vanuit een gegeven knoop en bezoekt daarbij alle kanten en knopen. Een aantal kanten wordt daarbij gelabeld met het label *discovery*: deze kanten vormen een opspannende boom van de graaf.
   Aanwijzing: maak gebruik van recursie, en van een geschikte labeling van de knopen.

4. (25 punt) Geef definities van de begrippen *standaard trie* (van een verzameling S van strings), *gecomprimeerde trie* (idem), *compacte trie* (idem) en *suffix trie* (van een string T).
   Maak één van de onderstaande twee deelopgaven naar keuze.

   (a) Gegeven is een standaard trie van een verzameling S van strings. Geef een algoritme in pseudocode dat nagaat of een string w in S voorkomt.

   (b) Gegeven is een suffix trie van de string T. Geef een algoritme in pseudocode dat nagaat of een patroon P voorkomt in T.