

# Tentamen Algoritmen en Datastructuren in C

maandag 4 juli 2011, 14 - 17 uur

1. (25 punt)

- (a) Geef een definitie van de stack als abstract datatype.
- (b) Laat zien hoe een stack in C geïmplementeerd kan worden met behulp van een array. Definieer daarbij de functies `push` en `pop`. `push` verdubbelt de omvang van de stack indien deze vol is, bv. met behulp van de functie `stackFull`.
- (c) Gegeven zijn twee stacks  $S_0$  en  $S_1$  en de bijbehorende functies `push_0`, `push_1`, `pop_0`, `pop_1`, `stackEmpty_0`, `stackEmpty_1` (deze laatste functies geven aan of de betreffende stack leeg is). Implementeer hiermee een queue, dwz. geef definities van `enqueue`, `dequeue`, `queueEmpty`. Je mag hierbij naar keuze C of pseudo-code gebruiken.

2. (25 punt)

- (a) Geef een `typedef` in C voor een gelinkte lijst met per knoop een getal van type `int`. Geef hierbij C-functies `addItem` en `removeFirstNode` om een getal aan een lijst toe te voegen, en de eerste knoop uit een lijst te verwijderen. Zorg ervoor dat vrijgekomen geheugen correct wordt teruggegeven.
- (b) Geef vervolgens een C-functie `voegGeordendIn` die een getal toevoegt aan een geordende lijst, zodanig dat de lijst geordend blijft.

Z.O.Z.

3. (25 punt)

- (a) Wanneer is een ongerichte graaf *samenhangend*?
- (b) Geef een algoritme in pseudocode dat, gegeven een ongerichte samenhangende graaf en twee knopen  $v$  en  $w$ , de lengte van een kortste pad tussen  $v$  en  $w$  bepaalt. De lengte van een pad is hier het aantal kanten van het pad.

*Aanwijzing.* Dit is een speciaal geval van het probleem van het vinden van een kortste pad in een ongerichte *gewogen* graaf: alle kanten hebben gewicht 1. Je kunt dus het idee van het algoritme van Dijkstra gebruiken, maar een eenvoudiger aanpak is om gebruik te maken van het idee van *breadth-first search*. Je mag gebruik maken van een queue of een priority queue.

4. (25 punt)

- (a) Geef een definitie van het begrip *suffix trie* van een string  $T$ . Geef ter illustratie de suffix trie van het woord ABRACADABRA.
- (b) Gegeven is de suffix trie van string  $T$ . Geef een algoritme in pseudocode dat nagaat of een patroon  $P$  voorkomt in  $T$ . Geef ook de tijdscomplexiteit van het algoritme, en beargumenteer deze.