

Datastructuren: Opdracht 1

Lijsten en Stacks

Deadline:
29 September 2013 om 23:59:59

Het doel van deze opdracht is het implementeren van drie verschillende soorten lijsten, een stack en twee sorteer opgaven. De genoemde datastructuren zijn besproken in de hoorcolleges.

1 Lijsten

De drie verschillende lijsten die gevraagd worden zijn: een enkel verbonden (singly linked) lijst, een dubbel verbonden (doubly linked) lijst en een circulaire variant. De circulaire variant mag zowel enkel als dubbel verbonden zijn. Een lijst bevat *tenminste* de volgende operaties :

- `get()`
Voor het opvragen van elementen in de lijst
- `insert()`
Voor het toevoegen van elementen aan de lijst
- `remove()`
Voor het verwijderen van elementen uit de lijst
- `length()`
Voor het opvragen van het aantal elementen in de lijst

Een lijst moet ook netjes aangemaakt en verwijderd kunnen worden. Naast de verplichte functies ben je vrij om andere functies te implementeren.

2 Stack

Met behulp van een lijst kan ook een stack datastructuur geïmplementeerd worden. Een `Stack` klasse bevat *tenminste* de volgende operaties:

- `push()`
meegegeven element wordt bovenste element van de stack
- `pop()`
teruggegeven element is van de bovenkant van de stack afgehaald
- `empty()`
teruggegeven waarde geeft aan of de stack leeg is

De stack moet ook netjes aangemaakt en verwijderd kunnen worden. Naast de verplichte functies ben je vrij om andere functies te implementeren.

3 Opgaven

1. Sorteert een lijst met behulp van een sorteermethode naar keuze (selection, quick, merge, ...)
2. Sorteert een stack met gebruik van twee extra stacks en een losse variabele.

4 Eisen

De opdracht wordt beoordeeld op werking, keuzes van de implementatie, netheid van programmeren en verdere navolging van de eisen die in deze opdracht worden gesteld. De implementatie moet gebruik maken van *templates*, *const-correctness* en *inheritance*. De code moet foutloos te compileren zijn onder g^{++1} . Er mag geen gebruik worden gemaakt van de C++ STL datastructuren en algoritmes².

Er is voor deze opdracht een raamwerk beschikbaar gesteld. Dit raamwerk is te vinden op de website van het vak. Gebruik van dit raamwerk is optioneel maar wordt wel aangeraden.

Inleveren Er moet een Makefile en een tekstbestand meegeleverd worden. Het tekstbestand bevat de namen en studentnummers van de makers, instructies voor het compileren en voor het draaien van de verschillende opdrachten, een duidelijke uitleg over de invoer en uitvoer hiervan en eventueel ander commentaar. De broncode, makefile en het tekstbestand moeten samen ingepakt worden in een archief, zoals zip/gzip/rar/tgz. Stuur geen executables mee!

De uitwerking dient uiterlijk op zondag 29 september om 23:59:59 via email opgestuurd te worden naar Jan van Rijn (j.n.van.rijn@liacs.leidenuniv.nl).

¹De g^{++} compiler die geïnstalleerd staat op de LIACS Linux werkplekken wordt als referentie gebruikt. Er hoort gecompileerd te worden met tenminste de `-Wall -Wextra` flags.

²De standaard `#includes` zoals `<iostream>`, `<cstdlib>`, `<cstdio>`, `<ctime>` zijn wel toegestaan. Bij twijfel, vraag het aan de assistenten.