UNIVERSITEIT GENT

Taak 2: L-vormige betegeling

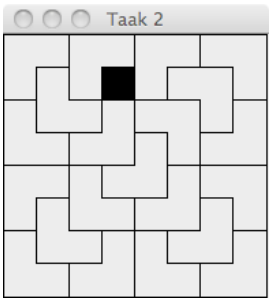
Algoritmen en Datastructuren I

Academiejaar 2010-2011

1 Probleem

Een **L-tegel** is een tegel bestaande uit drie vakjes in een L-vorm. Er wordt gevraagd een vierkant met één leeg vakje onder te verdelen in dergelijke L-tegels. De lengte van de zijde van het vierkant is steeds een **macht van twee**. Het lege vakje kan zich op om het even welke plaats bevinden en mag **niet** bedekt worden door een tegel. Alle andere vakjes in het vierkant moeten bedekt worden door **juist één tegel**. De tegels mogen **niet overlappen** en moeten zich **volledig** binnen het vierkant bevinden.

Een correcte onderverdeling in L-tegels ziet er bijvoorbeeld als volgt uit:



Er wordt gevraagd om een **recursief verdeel-en-heersalgoritme** te ontwerpen en te implementeren voor dit probleem. De nadruk zal bij deze opdracht vooral liggen op **snelheid** en **efficiëntie**, maar het spreekt voor zich dat je algoritme natuurlijk ook **correct** moet zijn.

2 Implementatie

Schrijf een klasse `MyLTiles` die de interface `LTiles` implementeert. Deze interface heeft slechts één methode:

```
int[][] tile(int k, int row, int col)
```

De parameter `k` geeft de **exponent** weer. Dit betekent dat de zijde van het vierkant een lengte van 2^k heeft (en dus geen lengte van k!). De parameters `row` en `col` geven de positie aan van het lege vakje (geteld vanaf 0). Het resultaat wordt geretourneerd in de vorm van een 2-dimensionele tabel met de volgende specificaties:

- De grootte van de tabel is $2^k * 2^k$.
- Indien `result` de naam is van de tabel, dan bevindt het lege vakje zich op positie `result[row][col]` en wordt aangegeven met -1.
- De tegels worden aangegeven met getallen van 0 tot en met $n - 1$, waarbij n het aantal gebruikte tegels is. Elk vakje in de tabel bevat het nummer van de tegel waartoe het behoort.
- Alle getallen van 0 tot en met $n - 1$ komen dus precies 3 keer voor in de tabel, terwijl -1 juist 1 keer voorkomt.

Voor het eerder gegeven voorbeeld ziet een dergelijke tabel er bijvoorbeeld als volgt uit:

20	20	11	11	15	15	6	6
20	0	11	**-1**	15	8	8	6
7	0	0	18	9	9	8	12
7	7	18	18	17	9	12	12
2	2	5	17	17	19	10	10
2	1	5	5	19	19	14	10
13	1	1	4	16	14	14	3
13	13	4	4	16	16	3	3

Merk op dat er geen volgorde opgelegd is voor de nummering. De volgende tabel zou voor datzelfde voorbeeld bvb. ook correct zijn:

8	8	17	17	0	0	10	10
8	12	17	**-1**	0	4	4	10
16	12	12	14	9	9	4	1
16	16	14	14	13	9	1	1
19	19	20	13	13	5	6	6
19	2	20	20	5	5	7	6
3	2	2	11	15	7	7	18
3	3	11	11	15	15	18	18

Met behulp van de klasse `TilePanel` is het mogelijk om je resultaat te visualiseren. Je doet dit door een nieuwe `TilePanel` aan te maken met het resultaat van de `tile` methode en dit in bijvoorbeeld een `JFrame` te plaatsen.

3 Verslag

In een **bondig** verslag van **1 à 5 bladzijden** beschrijf je duidelijk de werking van je algoritme. Vermeld ook alles wat je eventueel gedaan hebt om je algoritme te optimaliseren/versnellen. Bespreek ook de **tijdscomplexiteit** van je algoritme. Er worden voor deze taak **geen tijdsmetingen** gevraagd.

4 Nog enkele algemene opmerkingen

- Plaats je code **niet in een package-structuur** en voorzie je code van voldoende duidelijke **commentaar**.
- De klasse `MyLTiles` moet een **default-constructor** (zonder parameters) hebben. Dit wordt gecontroleerd m.b.v. de klasse `ConstructorTest`.
- Zeer belangrijk: **schrijf niets naar het scherm**, ook geen debuginformatie of foutmeldingen, niet naar stdout en niet naar stderr.
- Voor je verslag gebruik je de structuur van de gegeven **template** (`TemplateVerslag.odt`). Het is ook toegelaten dezelfde structuur over te nemen in LaTeX of in een andere tekstverwerker.

5 Indienen

Om in te dienen bundel je je verslag en code in een zip-bestand. Dit bestand bevat **enkel** een bestand `verslag.pdf` en je zelfgeschreven javabestanden. We verwachten dat je dit bestand indient voor **donderdag 31 maart 2011 om 14u30** via `http://indianio.ugent.be` bij het project: "Algoritmen en Datstructuren I: Taak 2".

Hiernaast dient er ook een **papieren versie** van het verslag ingediend te worden. Dit kan tot de deadline in bureau 40.09.110.016 (Bart) of 40.09.110.032 (Stéphanie) of ten laatste in het practicum van donderdag 31 maart 2011.

6 Belangrijke opmerking

Het project is **individueel** en dient dus door jou persoonlijk gemaakt te worden. Het is uiteraard toegestaan ideeën uit te wisselen maar het is ten strengste verboden code uit te wisselen, op welke manier dan ook. Het overnemen van code beschouwen we als fraude van beide betrokken partijen en zal in overeenstemming met het examenreglement behandeld worden.