

Faculteit Wiskunde & Informatica

Toets Datastructuren

Vrije Universiteit

18 mei 1998

tijdsduur : 1.5

uur

Opgave 1.

a) `public static int AANTAL_ELEMENTEN = 1000;`

`int[] rij = new int[AANTAL_ELEMENTEN];`

`// .....`

`// .....`

`// Alle AANTAL_ELEMENTEN elementen van rij hebben een waarde gekregen.`

In rij wordt nu gezocht of de waarde x voorkomt. Dit gebeurt met linear search. Van welke orde is dit zoekalgoritme?

Vervolgens wordt de rij gesorteerd in monotoon niet-dalende volgorde m.b.v. een eenvoudige sorteermethode (simple sort). Van welke orde is zo'n sorteermethode?

Nu de rij gesorteerd is, is het mogelijk om m.b.v. binary search te zoeken of een waarde x voorkomt. Van welke orde is binary search?

Geef met getallen aan hoeveel handelingen er ongeveer nodig zijn bij linear search, bij een eenvoudige sorteermethode en bij binary search op de variabele 'rij'.

- b) Leg uit hoe binary search werkt. Maak het idee duidelijk. Er wordt dus niet gevraagd om binary search te programmeren.
- c) Bespreek hoe een hash tabel met "seperate chaining" werkt. Gebruik en verduidelijk in je bespreking de termen "hash function", "collision", "key" en "data".

Opgave 2.

Gegeven is de onderstaande class voor knopen in een dubbelgeketende circulaire lijst (doubly linked circular list).

```
class Knoop {  
  
    int data;  
  
    Knoop prior, next;  
  
    public Knoop (int data) {  
        this(data, null, null);  
    }  
}
```

```
}
```

```
public Knoop (int data, Knoop p, Knoop n) {
    this.data = data;
    prior = p;
    next = n;
}
}
```

Programeer een class Lijst die m.b.v. de class Knoop een circulaire lijst implementeert. Deze class Lijst moet er voor zorgen dat ieder Lijst-object een referentie naar het eerste en naar het laatste element van de circulaire lijst heeft en het aantal knopen bijhoudt.

♀

In de class Lijst moeten de volgende methodes aanwezig zijn:

```
public void init ();
```

```
/* PRE -
```

```
    POST - de lijst is leeg.
```

```
*/
```

```
public void insert (int data);
```

```
/* PRE -
```

```
    POST - lijst-POST = lijst-PRE + knoop met inhoud data
```

```
*/
```

```
public boolean aanwezig (int data);
```

```
/* PRE -
```

```
    POST - TRUE: er is een knoop in lijst met inhoud data
```

```
        FALSE: er is geen knoop in lijst met inhoud data
```

```
*/
```

```
public void remove (int data);
```

```
/* PRE - het aantal knopen in lijst-PRE met inhoud data is > 0
```

```
    POST - het aantal knopen in lijst-POST met inhoud data is >= 0
```

```
*/
```

Verder moeten er een parameterloze constructor en de copy-constructor aanwezig zijn in de class Lijst.

Beoordelingstabel opgave | a | b | c | d | e | f | totaal

1		4		4		6						14
2		22										22

1998-05-18 Tentamen.txt

TOTAAL	36
--------	----

Het eindcijfer E wordt als volgt uit het totaal T verkregen : $E = T/4 + 1$.