

Examen Datastructuren en Algoritmen I

1. Wat is de tijdscomplexiteit (in Θ -notatie) van het algoritme `kleinsteVanGrootste` uit het volgende programmafragment, wanneer een $n \times m$ matrix als parameter opgegeven wordt? Verklaar.

```
static int grootste (int [] rij) {
    int gr = rij[0];
    for (int i = 1; i < rij.length; i++)
        if (rij[i] > gr) gr = rij[i];
    return gr;
}
static int kleinsteVanGrootste (int [] [] mat) {
    int kl = grootste (mat[0]);
    for (int i = 1; i < mat.length; i++) {
        int gr = grootste (mat[i]);
        if (gr < kl) kl = gr;
    }
    return kl;
}
```

2. Gegeven zijn de experimentele tijdsmetingen voor een programma dat een bepaald algoritme implementeert. Het programma werd uitgevoerd voor verschillende waarden van de probleemgrootte n . Volgende tabel geeft de meetresultaten in milliseconden. Wat is de tijdscomplexiteit (in Θ -notatie) van dit algoritme? Verklaar.

n	10	11	12	13	14	15
uitvoeringstijd (in ms)	1200	2460	5000	10040	20110	41000

5. Gegeven zijn twee strings, bijvoorbeeld strings die een DNA-sequentie voorstellen.

Gevraagd is te bepalen of de ene string een 'cyclische permutatie' van de andere is. Hiermee wordt bedoeld dat de ene string gevormd kan worden door de andere over een bepaald aantal posities te verschuiven, en daarbij de string als een cyclische structuur te beschouwen.

Bijvoorbeeld, de string ATGCAT is een cyclische permutatie van de string TGCATA.

Geef een algoritme voor dit probleem. Wat is de tijds- en geheugencomplexiteit van dit algoritme?

6. Beschouw de volgende Java-code :

```
static void genereer (int [] rij) {
    genereer (rij, rij.length-1);
}
static void genereer (int [] rij, int tot) {
    if (tot <= 0)
        nieuweOplossing (rij);
    else
        for (int i = tot; i >= 0; i--){
            verwissel (rij, i, tot);
            genereer (rij, tot-1);
        }
}
static void nieuweOplossing (int [] rij) {
    System.out.print "(" + rij[0];
    for (int i = 1; i < rij.length; i++)
        System.out.print "," + rij[i];
    System.out.println ("");
}
static void verwissel (int [] rij, int i, int j) {
    int tmp = rij[i]; rij[i] = rij[j]; rij[j] = tmp;
}
}
```

Wat is de output als de methode `genereer` opgeroepen wordt met de rij (8,3,5) als parameter?

-
7. Een weerkundig instituut krijgt dagelijks een bestand met de metingen van het ozon-gehalte in de lucht (gegeven als een reëel getal) op een aantal vastgelegde meetplaatsen (gegeven als string). Gegeven zijn de bestanden met de gegevens voor het jaar 2000, een bestand per dag; in elk bestand zijn de meetwaarden uit n meetplaatsen opgenomen. Er wordt gevraagd om te bepalen in welke meetplaatsen het ozon-gehalte op meer dan k dagen een bepaalde gegeven kritische waarde overschreed.

Geef een algoritme voor dit probleem, en geef aan hoe daarbij de standaard datatypes (zoals de Java-collections) kunnen gebruikt worden. Wat is de tijds- en geheugencomplexiteit van het algoritme?

10. Gevraagd wordt een gedeeltelijke Java-implementatie van hashtabellen met open hashing (ook hashing met gescheiden ketening genoemd) :

(a) Geef de datavelden voor het voorstellen van een hashtable.

(b) Implementeer een constructor die een lege hashtable aanmaakt, met gegeven tabelgrootte.

(c) Implementeer de bewerking boolean add (Object x) (zoals gedefinieerd in de interface *Set*).

Bij deze implementatie mag gebruik gemaakt worden van het ADT *List*, en een van zijn implementaties *ArrayList* en *LinkedList*, zoals die geleverd worden door het collections-framework van Java. (Zie bijlage uit Java API documentatie van *List*.)
