Opdracht 2 theoretische informatica

W. Oele

15 november 2024

Opgave 1

Leidt de recurrente betrekking af voor het aantal verbindingslijnstukken tussen n punten. Tussen twee punten is één verbindingslijnstuk aanwezig.

Opgave 2

De volgende functie berekent de faculteit van een natuurlijk getal $n \geq 0$:

```
int faculteit(int n > 0)
 if (n <= 1)
  return n
 else
  return n*faculteit(n - 1)
```

Maak een iteratieve versie van deze functie.

Opgave 3

Leidt het aantal stappen T_n af voor het verplaatsen van n schijven bij de torens van Hanoi, indien rechtstreekse verplaatsingen van toren A naar toren C verboden zijn. Elke schijf moet langs toren B.

Opgave 4

Bereken *alg_a(n)* en *alg_b(n)* voor $n = 1 \ldots 5$. Bereken de efficiëntie van algoritme *alg_a* en van algoritme *alg_b* in O-notatie:

1. 1. alg_a(n):resultaat
   2. **if** $n > 1$ **then**
   3. **return**$(alg\_a(n-1) + alg\_a(n-1))$
   4. **else return**(1)

2. 1. alg_b(n):resultaat
   2. **if** $n > 1$ **then**
   3. **return**$(2 \cdot alg\_b(n-1))$
   4. **else return**(1)

Opgave 5

Leidt een recurrente betrekking af voor een vermenigvuldiging van twee n-bit natuurlijke getallen x en y. Maak hiervan een recursief algoritme. Bepaal tevens de tijdcomplexiteit van dit algoritme. Wij mogen veronderstellen dat voor het betrokken processortype, het vermenigvuldigen en delen $O(n^2)$ instructies zijn. Daarentegen zijn optellen, aftrekken en 1-bits schuifacties $O(n)$ instructies. Het testen of een natuurlijk getal even-of oneven is, is zoals alle overige instructies atomair $O(1)$.

Opgave 6

Leidt een recurrente betrekking af voor de berekening van x^p, waarbij x een reëel getal en p een natuurlijk getal is van n bits. Maak hiervan een recursief algoritme. Bepaal de tijdcomplexiteit voor dit algoritme op dezelfde processortype als die uit de vorige opgave.

Opgave 7

7a Schrijf een functie met parameter n die de volgende berekening uitvoert:

$$\sum_{j=1}^{j=n} \frac{1}{j(j+1)}$$

7b Test je functie met verschillende waarden voor n.

7c Schrijf een efficiëntere versie van deze functie en bewijs dat de efficiëntere versie *altijd* hetzelfde antwoord oplevert als de oorspronkelijke versie.

7d Geef de herhalingsfrequentie en complexiteitsontwikkeling van beide functies.

Opgave 8

Een ingewikkelde vorm van recursie is de functie van *Ackermann* :

$$Ack(m,n) = \begin{cases} n+1 & \text{als } m = 0 \text{ en } n \geq 0 \\ Ack(m-1,1) & \text{als } m > 0 \text{ en } n = 0 \\ Ack(m-1, Ack(m, n-1)) & \text{als } m > 0 \text{ en } n > 0 \end{cases}$$

Toon aan dat $Ack(2,3) = 9$.