Prof. Andreas Goerdt
Frank Schädlich

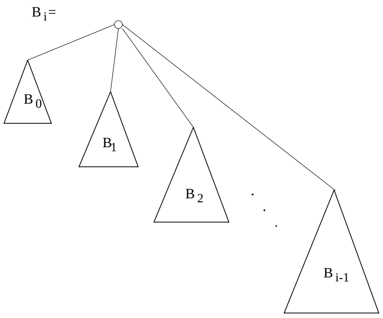
TU Chemnitz
Sommersemester 2003

Theoretische Informatik III
1. Übung

1. Aufgabe

Sei B_i der i-te binomiale Baum.

(a) Zeigen Sie für jedes $i > 0$, daß B_i folgende Struktur hat:



(b) Zeigen Sie, daß Tiefe$(B_i) = i$ für alle $i \geq 0$ gilt. Zeigen Sie weiter, daß B_i genau *einen* Knoten in Tiefe i hat.

(c) Zeigen Sie für die Anzahl Punkte $|B_i|$ in B_i: $|B_i| = 2^i$ für alle $i \geq 0$.

2. Aufgabe

Beweisen Sie:

(a) Für alle $k, n \in \mathbb{N}$ mit $n, k \geq 1$ ist $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$.

(b) Für alle $n \geq 0$ ist $\sum_{i=0}^{n} \binom{n}{i} = 2^n$.

(c) Zeigen Sie, daß für alle $k, l, n \in \mathbb{N}$, $k \leq l + n$ gilt:

$$\binom{l+n}{k} = \sum_{i=0}^{l} \binom{l}{i} \cdot \binom{n}{k-i} \ .$$

Dabei gilt $\binom{a}{b} = 0$ falls $a < b$ ist.
Hinweis: Arbeiten Sie mit der „kombinatorischen Interpretation" der Binomialkoeffizienten:

$$\begin{aligned} \binom{l+n}{k} &= \sharp\ k\text{-elementige Teilmengen einer } (l+n)\text{-elementigen Menge} \\ &= \sharp\ k\text{-elementige Teilmengen einer Menge,} \\ &\quad\ \text{die aus } l \text{ „roten" und } n \text{ „schwarzen" Elementen besteht.} \end{aligned}$$

3. Aufgabe

Sei F_i die i-te Fibonacci-Zahl ($F_0 = 0, F_1 = 1, F_i = F_{i-2} + F_{i-1}$ für alle $i > 1$). Zeigen Sie, daß für alle $i \in \mathbb{N}$ und $i > 1$ gilt:

$F_i \geq \varphi^{i-2}$,

wobei $\varphi = \frac{1+\sqrt{5}}{2}$ der goldene Schnitt ist.

4. Aufgabe

Zeigen Sie:

(a) Ist $g(n) \in O(f(n))$, so ist für jedes $c \geq 0$ die Funktion $c \cdot g(n) \in O(f(n))$.

(b) $f(n) \in O(f(n))$.

(c) Ist $g(n) \in O(f(n))$ und $h(n) \in O(g(n))$, so ist $h(n) \in O(f(n))$.

(d) Ist $h(n) \in O(f(n) + g(n))$, so ist $h(n) \in O(Max\{f(n), g(n)\})$.

(e) Ist $h(n) \in O(f(n))$ und $l(n) \in O(g(n))$, so ist $h(n) \cdot l(n) \in O(f(n) \cdot g(n))$.

5. Aufgabe

Gegeben sei ein ungerichteter Graph $G = (V, E)$ mit $V = \{v_1, \ldots, v_n\}$. Die Kosten der Kante $\{u, v\}$ betragen $w(\{u, v\})$. Ein Algorithmus zum Bestimmen eines minimalen Spannbaums T arbeitet mit einer Partition $\{V_1, V_2, \ldots, V_k\}$ von V. Wir benutzen die Notation $E_i = \{\{u, v\} : u \in V_i \text{ oder } v \in V_i\}$.
Folgender Pseudocode erzeugt den minimalen Spannbaum T:

```
T := ∅
for (jeder Knoten v_i ∈ V) do
    V_i := {v_i}
    E_i := {{v_i, v} ∈ E}
while (es gibt mehr als eine Menge V_i) do
    wähle beliebige Menge V_i
    extrahiere Kante {u, v} ∈ E_i mit den minimalen Kosten
    o.B.d.A. sei u ∈ V_i und v ∈ V_j
    if i ≠ j then
        T := T ∪ {{u, v}}
        V_i := V_i ⋃ V_j, lösche V_j aus der Partition
        E_i := E_i ⋃ E_j \ {{u, v}}
```

Beschreiben Sie eine Implementierung dieses Algorithmus unter Verwendung von binomialen Heaps für die Mengen E_i sowie einer geeigneten Union-Find-Struktur für die Partition von V und analysieren Sie die Laufzeit.