**Algorithmen zur Sequenzanalyse**

Wintersemester 2017/2018 — **Übungsblatt 6** — Prof. Dr. E. Ohlebusch,

Besprechung am 26.01.2018 — Institut für Theoretische Informatik

---

**Aufgabe 6.1**.

Berechnen Sie mit Hilfe von Algorithmus 27 im Skript das LCP-Array für den String $S = annasanannas\$$. Die Burrows-Wheeler-Transformierte von $S$ ist $\mathsf{BWT} = ssn\$nnannaaaa$.

**Aufgabe 6.2**.

Die Burrows-Wheeler-Transformationen der Strings $\mathsf{S}$ und $\mathsf{S}^{rev}$ sind jeweils in einem Wavelet-Baum gespeichert, siehe Abbildungen 1 und 2. Ermitteln Sie mittels Algorithmus 29 im Skript, wie oft der String $GTA$ in $\mathsf{S}$ vorkommt.

**Aufgabe 6.3**.

Erweitern Sie die Vorkommen (beginnend mit den gefundenen Vorkommen aus Aufgabe 2) jeweils mit einem Rückwärtssuchschritt um den Buchstaben $c$ und versuchen Sie mit einem Vorwärtssuchschritt das Muster auf der anderen Seite mit dem Watson-Crick-Komplement von $c$ zu erweitern. Wie lautet das längste Muster, dass Sie auf diese Weise finden können?
Die Watson-Crick-Paarungen sind $A$-$T$ und $C$-$G$.

**Aufgabe 6.4**.

Die Zeitkomplexität eines Suchschrittes mittels Algorithmus 29 im Skript wird durch die Zeitkomplexität des Aufrufes der Prozedur *getBounds* bestimmt. Die Ausführung von *getBounds* erfordert $O(\log \sigma)$ Zeit, wenn man den Wavelet-Baum benutzt (Algorithmus 28 im Skript). Geben Sie eine Implementierung von *getBounds* an, die mehr Platz als der Wavelet-Baum (nämlich $O(n\sigma)$ Bits) benötigt, sodass *getBounds* in $O(1)$ Zeit ausgeführt werden kann.

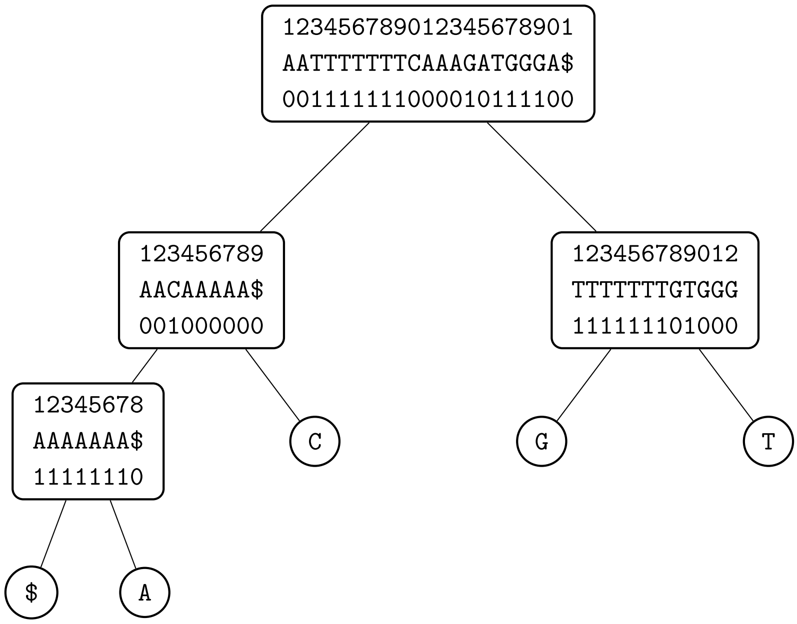


Abbildung 1: Wavelet-Baum von *AATTTTTTTCAAAGATGGGA$*

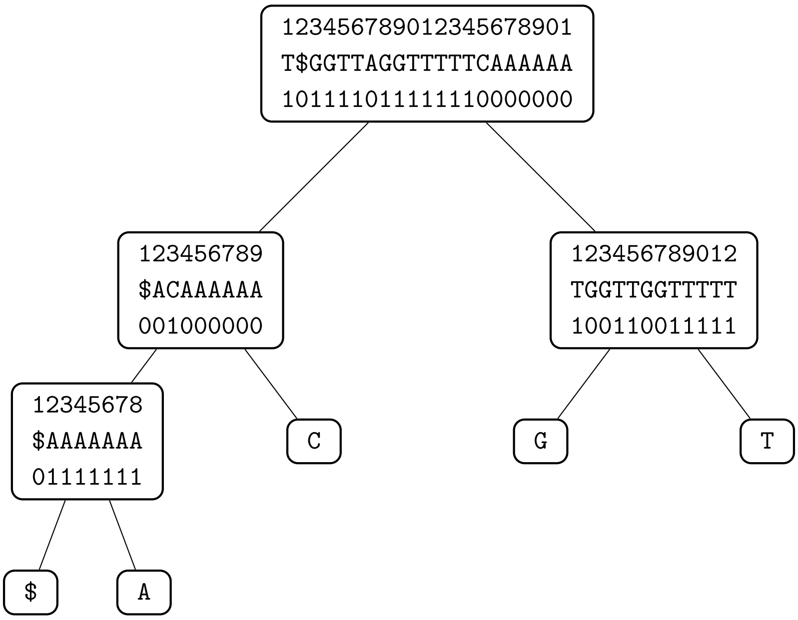


Abbildung 2: Wavelet-Baum von *T$GGTTAGGTTTTTCAAAAAA*