technische universität dortmund

Bioinformatik, Lehrstuhl XI
Fakultät für Informatik

Prof. Dr. Sven Rahmann

Wintersemester 2018/19

Übung zu Algorithmen auf Sequenzen
Blatt 2

Ausgabe: 18.10.2018 **Besprechung:** 25.10.2018

Aufgabe 2.1

Finden Sie eine möglichst effiziente Methode, um zu testen, ob $\text{popcount}(n) \leq 2$ ist. Konstruieren Sie daraus eine Methode, um $\text{popcount}(n)$ in einer Laufzeit proportional zur Anzahl der 1-Bits in n zu berechnen.

Aufgabe 2.2

Geben Sie eine exakte Definition der Funktionenklasse $o(g(n))$ an. Prüfen Sie, ob die folgenden Funktionen in $o(n)$ sind oder nicht. Mit log sei hier immer der Logarithmus zur Basis 2 gemeint.

1. $f_1(n) := n - C$ für konstantes $C > 0$
2. $f_2(n) := n/C$ für konstantes $C > 1$
3. $f_3(n) := n^{1-\varepsilon}$ für $0 < \varepsilon < 1$
4. $f_4(n) := n^{1-\varepsilon} + (\sqrt{n}\log n)^{\varepsilon}$ für $0 < \varepsilon < 1$
5. $f_5(n) := n/\log\log n$
6. $f_6(n) := 2^{\log n}$
7. $f_7(n) := 3^{\log n}$

Aufgabe 2.3

Nehmen Sie realistischerweise an, dass eine CPU Wörter der Bitlängen 8, 16, 32 und 64 effizient verarbeiten kann (Typen byte, short, int, long; jeweils unsigned). Wie groß sollte die Superblockgröße S der Rank-Datenstruktur jeweils gewählt werden für Bitsequenzen der Längen $n = 10^k$ für $k \in 3, 4, \ldots, 10$? Wie viele Bits benötigt die rank-Datenstruktur dann jeweils für eine optimale Wahl von S?

Aufgabe 2.4

Erinnern Sie sich an einen Beweis der geometrischen Reihe $\sum_{j=0}^{\infty} p^j = \frac{1}{1-p}$ für $0 \leq p < 1$. Berechnen Sie $\sum_{k=0}^{\infty} (k+2)(k+1)\, p^k$ für $0 \leq p < 1$.

Aufgabe 2.5

Analysieren Sie die erwartete Laufzeit der naiven Mustersuche, wenn die Buchstaben des Alphabets mit unterschiedlichen Wahrscheinlichkeiten vorkommen. Sei $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$. Die Wahrscheinlichkeit für den Buchstaben σ_i sei $0 \leq p_i \leq 1$ an jeder Stelle von Muster und Text, unabhängig von den anderen Stellen. Natürlich gelte $\sum_{i=1}^{k} p_i = 1$.