Kursarbeit Informatik 12 Name:_____
04.05.2021

1. Gegeben ist die für natürliche Zahlen $n \geq 0$ definierte Fibonacchi-Folge {f(n)}:

   Rekursionsanfang: **f(0) = 0**
   f(1) = 1

   Rekursionsvorschrift: **f(n) = f(n – 1) + f(n – 2)** , $n \geq 2$

   Schreibe einen Python-Quelltext mit rekursivem Funktionsaufruf, um nach Eingabe einer natürlichen Zahl n die Fibonacchi-Folge für alle natürlichen Zahlen aus dem Bereich 0, ... , n zu berechnen und auszugeben.

2. Der Algorithmus **MergeSort**

   Gegeben ist eine Liste **a** mit den **n** Komponenten **a[0], ... , a[n-1]** , für die die Ordnungsrelationen <, >, ≤, ≥ erklärt sind. Die Liste ist aufsteigend zu sortieren, so daß gilt: **a[0] ≤ a[1] ≤ . . . . . ≤ a[n-1]**

   a) Formuliere die Schritte, gemäß denen der Algorithmus MergeSort vorgeht, in Worten; an welcher Stelle zeigt sich die rekursive Definition des Algorithmus?

   b) Schreibe den Quellcode der Funktion **sort** als Python-Programm; erläutere auch, was die Funktionen **sort(a,l,r)** und **merge(a,l,m,r)** bewirken.

      Wie lautet der Aufruf, um eine Liste **a** mit **n** Datenelementen zu sortieren?

   c) Führe den Sortieralgorithmus MergeSort exemplarisch durch für die aus 8 Komponenten bestehende Liste

      [7, 5, 8, 1, 3, 6, 5, 2] .

      Notiere auch die jeweiligen Aufrufe von **sort** und **merge** (wahlweise in einem gesonderten Baumdiagramm).

      Querformat!

   d) Der Aufwand (Anzahl elementarer Operationen wie Vergleiche, Additionen, Wertzuweisungen) zum Sortieren von n Datenelementen werde mit A(n) bezeichnet.

      α) Begründe die Funktionalgleichung und die Anfangsbedingung, die sich für die Funktion A(n) formulieren lassen.

      β) Gib einen Funktionsterm für A(n) an und bestätige, daß dieser die Anfangsbedingung und die Funktionalgleichung erfüllt.

      γ) Über die Speicherkomplexität gibt die Anzahl f(n) der Aufrufe der Funktion sort Auskunft; motiviere für $n = 2^k$, $k \in \{0, 1, 2, ...\}$, einen Term für f(n) und zeige damit, daß die Speicherkomplexität gegenüber dem in β) ermittelten Aufwand A(n) zu vernachlässigen ist.
      (Hinweis: bei dem Beispiel aus c) gilt n = 8, also k = 3)

3. Der Algorithmus SelectionSort (Sortieren durch direkte Auswahl) hat bekanntlich quadratische Komplexität, d. h. für dessen Aufwand $\mathbf{A_S}$ gilt: $\mathbf{A_S \sim n^2}$. Der Aufwand $\mathbf{A_M}$ für MergeSort mit linear-logarithmischer Komplexität ist proportional zu $\mathbf{n \cdot \log_2(n)}$.

   Berechne die Quotienten

   $\mathbf{A_S(1\,000\,000)/A_S(1000)}$ und $\mathbf{A_M(1\,000\,000)/A_M(1000)}$.

   Bestimme daraus den Zeitbedarf, den SelectionSort und MergeSort jeweils benötigen, um eine Liste mit 1 Million Datenelementen zu sortieren, wenn das Sortieren von 1000 Elementen jeweils 1 ms erfordert.