Name: | Vorname: | Matr.–Nr.:

Technische Universität München
Fakultät für Informatik, I-16
Dr. Stefanie Demirci

SoSe 2017
7. Juni 2017

Probeklausur zu Algorithmen und Datenstrukturen

Allgemeine Hinweise

- Die Angabe besteht aus 2 Seiten. Stellen Sie sicher, dass Ihr Exemplar vollständig ist, und **prüfen Sie alle Seiten auf Aufgaben!**
- Sie haben 45 Minuten Zeit für die Bearbeitung der Aufgaben, von 09:45 bis 10:30 Uhr.

Aufgabe 1 Rechnen mit Landau-Symbolen (4 Punkte)

Untersuchen Sie, ob folgende Behauptung wahr ist und zeigen Sie Ihre Aussage.

$$\frac{1}{2}n^2 - 3n + 4 = O(n^3)$$

Aufgabe 2 Boolesche Logik (4 Punkte)

Zeigen Sie, dass der logische Ausdruck $a \Rightarrow b$ identisch ist mit $\neg a \vee b$.

Aufgabe 3 Zahldarstellung (4 Punkte)

Konvertieren Sie die Zahl 7_{10} in eine 4-bit Binärzahl. Berechnen Sie dann das 2-Komplement dieser 4-bit Binärzahl. Warum lässt sich das entstandene Bitmuster in der 2-Komplement-Darstellung als negative Zahl interpretieren?

Aufgabe 4 Sequentielle Liste **(2 Punkte)**

Gegeben sei ein Feld A implementiert als sequentielle Liste (Array). Was ist die Komplexität der Einfüge-Operation `insert` und der Zugriffs-Operation `elementAt`?

Aufgabe 5 Stack **(3 Punkte)**

Nennen Sie kurz *eine* Methode, um den abstrakten Datentyp Stack mittels elementarer Datenstrukturen zu implementieren. Was sind dann jeweils die Komplexitäten der Operation `push` und `pop`?

Aufgabe 6 Fibonacci-Folge, Verifikation **(6 Punkte)**

Die Fibonacci-Folge $(f_n)_{n\in\mathbb{N}}$ ist rekursiv definiert als:

$$\begin{aligned} f_1 &= f_2 &&= 1 \\ f_n &= f_{n-1} + f_{n-2} \end{aligned}$$

Nachfolgend ist die aus der Vorlesung bekannte iterative Implementierung nach dem Prinzip der dynamischen Programmierung angegeben als Funktion **fib**(n).

fib(n) :
    $fib =$ leeres Feld;
    $fib[1] = 1$; $fib[2] = 1$; $k = 3$;
    while $(k \leq n)$ {
        $fib[k] = fib[k-1] + fib[k-2]$;
        $k = k + 1$;
    }
    return $fib[n]$;

a) Geben Sie jeweils eine geeignete Vor- und Nachbedingung der Funktion **fib**(n) an!

b) Geben Sie eine geeignete Schleifeninvariante P für die **while** $(k \leq n)$ { ... } Schleife an!

c) Welche drei Bedingungen müssen überprüft werden, damit die Korrektheit der **while** Schleife mittels der Invariante P gezeigt werden kann? Zeigen Sie, dass diese drei Bedingungen in diesem Fall erfüllt sind!

Aufgabe 7 Sortieren **(8 Punkte)**

a) Geben Sie einen Sortier-Algorithmus an, der im Mittel mit Komplexität $O(n \log n)$ sortiert! Nach welchem Algorithmen-Muster wurde dieser Algorithmus entworfen?

b) Gegeben sei eine bereits sortierte Liste von natürlichen Zahlen der Länge n. In diese Liste soll nun ein weiteres Element einsortiert werden. Schlagen Sie einen möglichst effizienten Algorithmus für diese Aufgabe vor und geben Sie dessen Komplexität an.

c) Skizzieren Sie (durch Angabe des Zustandes nach jedem Durchlauf der Hauptschleife), wie der *Selection Sort* die folgende Zahlenreihe sortieren würde!

5 , 1 , 9 , 3 , 8 , 6 , 4 , 7