

5. Übungsblatt

Ziel: Auseinandersetzung mit Sortieralgorithmen.

1. Aufgabe (10 Punkte)

Nehmen Sie an, wir haben eine Reihe von Prozessen, die mit Prioritäten zwischen 1 und 100 abgearbeitet werden müssen.

$$[(p_1, \text{priority}_1), (p_2, \text{priority}_2), \dots, (p_n, \text{priority}_n)]$$

Simulieren Sie unter Verwendung einer binären *Heap*-Struktur (siehe Vorlesungsfolien) eine Warteschlange, in der die Prozesse nach Priorität abgearbeitet werden (*Priority Queue*).

Eine Prioritätswarteschlange speichert Objekte nach einer Prioritätseigenschaft und entfernt immer als erstes das Objekt mit der höchsten Priorität.

Schreiben Sie dafür folgende vier Funktionen:

```
def buildPriorityQueue( processList )
```

```
    """ Die processList wird in eine Heap-Struktur transformiert """
```

```
def insert( priorityQueue, newProcess )
```

```
    """ Ein neuer Prozess wird in dem priorityQueue eingefügt """
```

```
def isEmpty( priorityQueue )
```

```
    """ gibt einen Wahrheitswert als Rückgabewert, je nachdem, ob die  
    Prioritätswarteschlange leer ist oder nicht """
```

```
def removeProcess( priorityQueue )
```

```
    """ Der Prozess mit der höchsten Priorität wird aus der priorityQueue  
    entfernt und als Ergebnis der Funktion zurückgegeben """
```

Schreiben Sie eine Test-Funktion, die zufällig Prozesse einfügt oder entfernt. Die Prioritäten der Prozesse sollen auch zufällig ausgewählt werden.

Wenn die Prioritätswarteschlange leer ist, soll die **removeProcess**-Funktion den Wert **None** zurückgeben.

2. Aufgabe (5 Punkte)

a) Schreiben Sie eine Funktion, die aus einem beliebigen Text die Worte, die sich in den letzten drei Buchstaben reimen, in einer Liste von Gruppenworten organisiert.

Beispiel:

Eingabe: "Klaus baut ein Haus und klaut vom Nachbarn Strom."

Ausgabe: [[Klaus, Haus], [baut, klaut], ['ein'], ['und'], ['vom'], ...]

b) Schreiben Sie eine zweite Funktion, die mit linearem Aufwand die Wörter innerhalb der verschiedenen Gruppen sortiert.

Wir wissen noch dazu, dass die Worte eine maximale Länge von 32 Buchstaben haben können.

3. Aufgabe (5 Punkte)

Implementieren Sie den iterativen Algorithmus für die Lösung des Hanoi-Turm-Problems, das in der Vorlesung besprochen worden ist.

4. Aufgabe (4 Punkte)

Schreiben Sie eine Variante des Algorithmus von Erathostenes, der anstatt ein Array von booleschen Werten ein Array von Bytes verwendet. Die Bits der Zahlen sollen für das Markieren der Primzahlen verwendet werden. Auf diese Art und Weise kann der Wertebereich der suchenden Primzahlen um einem Faktor von 8 erhöht werden.

Weil für das Speichern eines booleschen Wert ein Byte in den meisten Programmiersprachen verwendet wird, hätten wir mit unserem Algorithmus eine echte konstante Verbesserung de Speicherverbrauchs.

5. Aufgabe (5 Punkte)

Schreibe eine Python-Funktion, die mit linearem Aufwand bezüglich der Anzahl von Pixeln in einem Schwarz/Weiss-Bild das grösste schwarze Quadrat innerhalb des Bildes findet. Das Bild wird mit einer Matrix aus Nullen und Einsen dargestellt, wobei 0 für weiss und 1 für schwarz steht.

Das Ergebnis soll in einem Tupel aus zwei Tupeln zurückgegeben werden, die der oberen linken und die unteren rechten Koordinate des Quadrates entspricht.

Beispiel:

```
0 0 1 1 0 1 1 1 0 0 0 0
0 0 1 0 1 0 1 1 1 1 1 0
0 0 0 0 0 1 1 1 1 1 1 0 => ((1,6), (4,9))
0 0 0 0 0 1 1 1 1 1 1 0
0 0 0 0 0 1 1 1 1 1 1 0
0 1 0 1 0 1 0 1 0 1 0 1
0 1 1 1 1 0 0 1 1 1 1 0
```

6. Aufgabe

Fangen Sie an, sich für die Zwischenklausur zu vorbereiten und wenn Sie sicher sind, dass Sie die Zwischenklausur schreiben möchten, melden Sie sich bis spätestens Montag den 28. Mai dafür an (siehe unsere Veranstaltungsseite).

Wichtige Hinweise:

- 1) Verwenden Sie geeignete Namen für alle Variablen und Funktionen, die den Inhalt der Variablen oder die Semantik der Funktionen darstellen.
- 2) Verwenden Sie vorgegebene Funktionsnamen, falls diese angegeben werden.
- 3) Kommentieren Sie Ihre Programme.

- 4) Verwenden Sie geeignete Hilfsvariablen und Hilfsfunktionen in Ihren Programmen.
- 5) Löschen Sie alle Programmzeilen und Variablen, die nicht verwendet werden.
- 6) Schreiben Sie getrennte Test-Funktionen für alle 5 Aufgaben für Ihren Tutor.