Divide&Conquer-Algorithmen lassen sich gut als rekursive Algorithmen darstellen. Das Prinzip eines rekursiven Algorithmus beruht darauf, ein schwieriges Problem in ein oder mehrere einfachere Probleme aufzuteilen. Man löst dann die einfacheren Probleme und kombiniert daraus die Lösung für das schwierige Originalproblem.

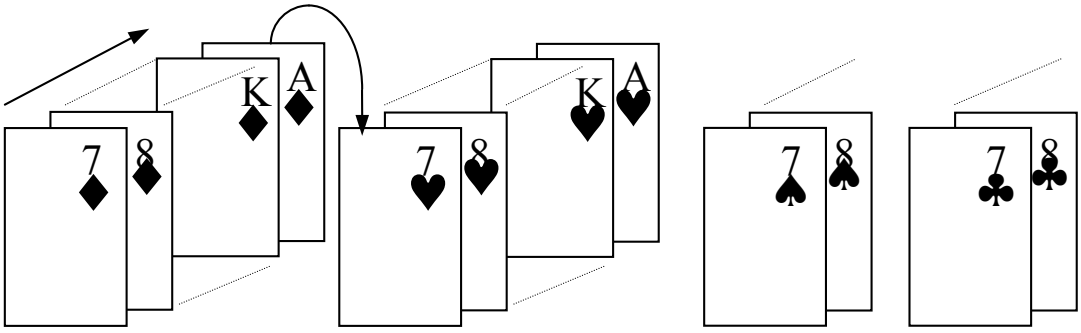
Problem Sortiere eine Folge von Objekten nach dem Divide&Conquer-Algorithmus ***MERGESORT***.

Aufgabe 1 (Experiment)

Ziel: Sortierung eines Kartenspiels

Sortierfolge nach Werten: 7 8 9 10 Bube Dame König As
nach Farben: Karo (♦) Herz (♥) Piek (♠) Kreuz (♣)

Die niedrigste Karte ist also die *Karo 7*, die höchste das *Kreuz As*.
Also: zum Beispiel liegt *Karo As* bei aufsteigender Sortierung vor *Herz 7*.



Regeln für das Experiment

1. **Teilen.** Erhalten Sie **<u>einen</u> Stapel**, dann ***teilen*** Sie diesen **in <u>zwei</u> gleichgroße Stapel** und

   Herrschen. ... geben Sie jeden Stapel an eine andere Person weiter, damit er diesen sortiert.

2. **Mischen.** Erhalten Sie **<u>zwei</u> Stapel** <u>zurück</u>, so ***mischen*** Sie diese wie folgt **zu <u>einem</u> neuen Stapel**: legen Sie die beiden Stapel offen vor sich hin, und entnehmen Sie entsprechend der obigen Sortierfolge jeweils von einem der beiden Stapel die **kleinste Karte** und stecken diese hinter den neuen Stapel. Sind beide Stapel leer, dann geben Sie den gemischten Stapel an denjenigen zurück, von dem Sie vorher einen Stapel bekommen haben.

3. **Ende.** Erhalten Sie einen Stapel mit **zwei Karten**, so sortieren Sie diesen und geben ihn zurück.

Aufgabe 2 (MERGESORT)

Formulieren Sie MERGESORT(A, p, r) rekursiv als Divide&Conquer-Algorithmus. Unterscheiden Sie in Ihrem Algorithmus deutlich die drei Schritte *Divide*, *Conquer* und *Combine*.

Hinweise: A = (A[1], A[2], ..., A[n]) ist die zu sortierende Folge, p ist der Index der ersten Elements und r der Index des letzten Elements der Folge A.

Gegeben ist eine Prozedur MERGE(A, p, q, r), die zwei Folgen (A[p], ..., A[q]) und (A[q+1], ..., A[r]) zu einer sortierten Folge (A[p], ..., A[r]) mischt.

Schreiben Sie MERGESORT:

Aufgabe 3 (Analyse von MERGESORT)

a) Identifizieren Sie in Ihrem Programm genau die Schritte *Divide*, *Conquer* und *Combine* und überlegen Sie sich für jeden dieser Schritte die Zeitkomplexität.

b) Stellen Sie mit den Überlegungen aus a) eine Differenzgleichung für MERGESORT auf.

c) Versuchen Sie die Differenzgleichung mit Hilfe des in der Vorlesung besprochenen Mastertheorems zu lösen.

Aufgabe 4

a) Gegeben ist ein Algorithmus, der sich selbst viermal rekursiv aufruft. In jedem der vier rekursiven Aufrufe hat das Datenfeld, das als Parameter übergeben wird, die Größe n/2; das Datenfeld wird also jeweils halbiert. Für die Zusammenführung der vier rekursiven Lösungen wird $O(n^2)$ Zeit benötigt.

1. Stellen Sie die **Differenzgleichung** des Mastertheorems T(n) für den oben beschriebenen Algorithmus auf und **erklären** Sie die Gleichung.
2. Welche asymptotische Laufzeit **T(n)** hat der Algorithmus?

Hinweis zum Mastertheorem:

$$T(n) = \begin{cases} O\left(n^{\log_b a}\right), & \text{für } a > b^k \\ O\left(n^k \cdot \log n\right), & \text{für } a = b^k \\ O\left(n^k\right), & \text{für } a < b^k \end{cases}$$

b) Gegeben sind zwei Algorithmen A1 und A2 mit den Laufzeiten

$T_{A1}(n) = b \cdot n^c$ und

$T_{A2}(n) = b \cdot c^n$,

mit den Konstanten b und c.

Erklären und begründen Sie, welcher der beiden Algorithmen sich mit Hilfe des Power-Tests experimentell analysieren lässt.