6. Übungsblatt	Grundlagen der Theoretischen Informatik III	SS 2012
Die Aufgaben werden am FR, 29.6., besprochen.		

1. Aufgabe: *(6+2 Punkte)*

`Mergesort` kann in der Praxis beschleunigt werden, indem für kurze Eingaben auf einen einfachen Sortieralgorithmus `Simplesort` umgeschaltet wird, mit asymptotisch schlechterer Laufzeit, aber kleineren Konstanten in der Laufzeitabschätzungsfunktion.

Beide Sortieralgorithmen erhalten als Eingabe ein Feld A von Elementen sowie einen linken Index ℓ und einen rechten Index r und bewirken, dass nach ihrer Ausführung $A[i] \leq A[i+1]$ für alle $i = \ell, \ldots, r-1$ gilt.

Wir möchte kurz eine Variante von `Mergesort`(A, ℓ, r) angeben:

1. IF $r - \ell = 0$ THEN return
2. $m = \lfloor (r-\ell)/2 \rfloor$
3. `Mergesort`(A, ℓ, m)
4. `Mergesort`$(A, m+1, r)$
5. `Merge`(A, ℓ, m, r)

Hierbei geht die Hilfsprozedur `Merge`(A, ℓ, m, r) davon aus, dass $A[i] \leq A[i+1]$ für alle $i = \ell, \ldots, m-1$ und für alle $i = m+1, \ldots, r-1$ gilt. Dieses "Mischen" bereits sortierter Felder geht in Linearzeit, ist jedoch für die Gesamtlaufzeit wesentlich.

Für unser Beispiel betrage die Laufzeit für die Sortierung von $n = r - \ell + 1$ Elementen:

- $n(n-1)/2$ für `Simplesort`(A, ℓ, r) und
- $7n$ für `Merge`(A, ℓ, m, r) für beliebige m zwischen ℓ und r.

Der Zeitbedarf aller übrigen Teile des folgenden Algorithmus `FastMergesort`$_L(A, \ell, r)$ sollen vernachlässigt werden.

1. IF $r - \ell \leq L$ THEN `Simplesort`(A, ℓ, r)
2. $m = \lfloor (r-\ell)/2 \rfloor$
3. `Mergesort`(A, ℓ, m)
4. `Mergesort`$(A, m+1, r)$
5. `Merge`(A, ℓ, m, r)

`Mergesort`(A, ℓ, r) und `FastMergesort`$_L(A, \ell, r)$ unterscheiden sich also nur in der ersten Zeile. Diskutieren Sie:

1. Welcher Wert ist optimal für L?
2. Wie groß ist der zeitliche Gewinn gegenüber $L = 0$?
   (Dann verhielte sich `FastMergesort`$_L(A, \ell, r)$ genauso wie `Mergesort`(A, ℓ, r) .)

2. Aufgabe: (*5 Punkte)*
In der Vorlesung wurde ein dynamisches Programm zur Lösung des Rucksackproblems beschrieben, das gutes Verhalten zeigte, wenn die Gewichtsschranke nur wenige Bits lang ist. Beschreiben Sie möglichst genau eine Lösung des Rucksackproblems, die "schnell" wäre, wenn umgekehrt die Profitschranke nur wenige Bits lang ist.