

1. SORTIERALGORITHMUS: MERGESORT

1.1. MOTIVATION

Wir entschieden uns für einen rekursiven Mergesort. Durch das Implementieren des Mergesorts werden folgende Aspekte geübt: Implementieren (und Debuggen) von rekursiven Algorithmen in Hoch- und Assemblersprachen, Allokieren von Speicherplatz und anschließendes Speichern von Daten auf dem Heap, und das Zusammenspiel zwischen Stack und Heap.

1.2. FORMALE ANALYSE DES MERGESORTS

1.2.1. Ablauf des Mergesorts in Pseudosprache

```
rekursiver_merge (Liste a)
  initialisiere den betrachteten Wert der linken Liste
  initialisiere den betrachteten Wert der rechten Liste
  für(k <= Listengröße)
    wenn (Anzahl Elemente Liste > 1)
      halbiere Liste;
      rekursiver_merge(linkte Hälfte der Liste);
      rekursiver_merge(rechte Hälfte der Liste);
      merge_beide_listen(linkte Hälfte, rechte Hälfte);
  merge_beide_listen(Liste linke Hälfte, Liste rechte Hälfte)
  für(k<= Listengröße)
    kopiere gesamten ursprünglichen Listeninhalt in temporäres Array;
  für(k<= Listengröße)
    wenn(1. Hälfte bereits gemerged) { kopiere den Rest der 2. Hälfte aus temporärem Array ins Zielarray }
    ansonsten wenn(2. Hälfte bereits gemerged) { kopiere den Rest der 1. Hälfte aus temporärem Array ins Zielarray }
    ansonsten { vergleiche die beiden Werte und kopiere den Kleineren ins Zielarray }
```

Code 1: Rekursiver Mergesort in Pseudosprache

1.2.2. Komplexität des Mergesort Algorithmus

Der Mergesort gehört zur Klasse der effizientesten bekannten Vergleichsalgorithmen. Aus Platzgründen wird auf den vollständigen Beweis verzichtet. Für weitere Informationen verweisen wir auf Sedgewick & Wayne „Algorithmen- Algorithmen & Datenstrukturen“ (Sedgewick & Wayne, 2014)¹: $O(N) = N * \log(N)$

2. ZUFALLSZAHLGENERATOR: KISS

2.1. MOTIVATION

Der Zufallszahlengenerator liefert durch die Addition von 3 Pseudozufallszahlen laut Greg Rose, Mitarbeiter des Forschungszentrums für Mobilfunkkommunikation von Qualcomm, ausreichend zufällig gestreute Zahlen für Simulationen², weswegen wir uns für diesen Algorithmus entschieden.

¹ Sedgewick, R., & Wayne, K. (2014). *Algorithmen - Algorithmen und Datenstrukturen*. S. 294 ff. Pearson Deutschland GmbH.

² Rose, G. (2011). *KISS: A Bit Too Simple*. Qualcomm Incorporated.

2.2. ANALYSE

2.2.1. Ablauf der Zufallszahlengenerierung (lt. Wikipedia)³

Beim KISS Zufallszahlengenerator werden 3 verschiedene Zahlen durch folgende Algorithmen generiert:

- **Linearer Kongruenzgenerator:** Zufallszahl1 = seed * Konst1 + Konst2
- **XOR Shifting Algorithmus:** Berechnen von Zufallszahl durch 3x Verschieben einer Zahl um Z Stellen, bilden des XOR mit der Zahl nach der Verschiebung und Zuweisung der Zahl als Ausgangspunkt für die nächste Iteration
- **Multiply-with-Carry Algorithmus:** Berechnen der Zufallszahl durch Multiplikation von 64- Bit mit 32-Bit Zahl, Addition mit Konstante, Verschiebung um 32 Bit nach rechts und casten als 32 Bit Zahl.

2.2.2. Praktischen Umsetzung des KISS Zufallszahlengenerators

Für das Seed wurde folgende Formel angewandt: $\text{seed} = \text{Adresse}_{\text{Stackpointer}} * t_{\text{Programmausführung}} / N_{\text{Eingabe}}$

Außerdem verzichteten wir aus Gründen der Einfachheit beim Multiply-with-Carry Algorithmus auf die Verwendung einer 64 Bit Zahl.

3. ZUSÄTZLICHE FEATURES

3.1. LESEN AUS EINER DATEI

Das Programm ist in der Lage, einen Input von einem File zu lesen. Die Input Datei muss *mergesort_recursive_input.txt* heißen und sich in **C:\assembler** befinden. Die Gleitkommazahl muss sich in einem validen Format befinden & die Datei muss < 12.500 Bytes sein. Eine valide Zahl genügt diesem regulären Ausdruck:

`([0-9A-F]{8}) (,[0-9A-F]{8})*.`

3.2. SCHREIBEN IN EINE DATEI

Der Output des Programms wird in *C:\assembler\mergesort_recursive_output.txt* geschrieben. Die Zahlen befinden sich in der Datei als ASCII Zeichen im Big Endian Format (von rechts nach links lesen).

3.3. ERRORHANDLING

- Negative Listenmengen werden abgewiesen und das Programm wird neu gestartet
- Semantische Prüfung der Eingabe: Der Mindestwert muss kleiner sein als der Maximalwert des gewünschten Intervalls, sonst wird das Programm neu gestartet
- Falls kein Input File besteht, obwohl es ausgewählt wurde, wird das Programm neu gestartet
- Es werden nur Zahlen von 0-9 und Großbuchstaben von A-F akzeptiert, die Zahlen müssen mit einem „ , “ getrennt sein und das Terminierungssymbol am Ende der Datei ist ein „ . “
- Bestehende Output-Datei wird geöffnet, falls nicht vorhanden, wird sie erstellt

3.4. VERWENDUNG DES HEAPS

Die generierten Zufallszahlen werden als Arrays auf den Heap abgelegt. Anschließend werden sie auf ein temporäres Array kopiert und von dort sortiert auf das Eingangsarray geschrieben.

³[https://de.wikipedia.org/wiki/KISS_\(Zufallszahlengenerator\)](https://de.wikipedia.org/wiki/KISS_(Zufallszahlengenerator))