

Algorithmen und Datenstrukturen SS 2020 – Praktikum 1

Zur Bearbeitung zwischen 8. Juni - 12. Juni (KW 24)

Wir betrachten das Problem „Maximale Teilsumme“ von Übungsblatt 1. Lesen Sie zur Auffrischung erneut Aufgabe 2 von Übungsblatt 1 (ab jetzt **Ü1-A2**) sowie die bereitgestellten Musterlösungen.

Aufgabe 1 (Maximale Teilsumme in $O(n^2)$)

Die Eingabe enthält einen Testfall pro Zeile. Jeder Testfall beginnt mit einer Zahl $0 \leq n \leq 10^4$, der Länge eines Arrays $A[0, \dots, n-1]$. Es folgen die n Zahlen $A[0], \dots, A[n-1]$. Pro Testfall ist eine Zeile mit der Zahl $s = \max_{0 \leq i \leq j+1 \leq n} \sum_{k=i}^j A[k]$ auszugeben.

Listing 1: sample.in

```
1 4 1 2 3 4
2 4 2 -1 3 -1
3 5 1 -1 2 3 -4
4 7 2 -3 5 -1 -2 4 1
```

Listing 2: sample.ans

```
1 10
2 4
3 5
4 7
```

- (a) Angenommen Sie verwenden den Ansatz aus **Ü1-A2-(a)**, der eine Laufzeit von $O(n^3)$ hat. Überschlagen Sie: Wie viele Sekunden würde das Programm für einen Testfall mit $n = 10^4$ benötigen, wenn es auf einem Computer ausgeführt wird, der 10^9 Rechenschritte pro Sekunde (1 GHz) erledigen kann.
- (b) Führen Sie die selbe Überschlagsrechnung bzgl. **Ü1-A2-(b)** durch. Implementieren Sie diesen Ansatz und reichen Sie ihn für das Problem **p1-a1-maxsum** im DOMjudge System ein.

Aufgabe 2 (Maximale Teilsumme in $O(n \log n)$)

Die größte erlaubte Eingabegröße beträgt nun $n = 10^6$. Implementieren Sie den Ansatz **Ü1-A2-(c)** und reichen Sie Ihre Lösung für das Problem **p1-a2-maxsum-rec** ein.

Hinweis: Den Materialien liegt eine Datei `startHilfe-a3.cpp` bei, die Sie als Ausgangspunkt für Ihre Implementierung nutzen können.

Den Materialien liegt zusätzlich zu `sample.in` und `sample.ans` auch ein zweites Eingabepaar `special.in` / `special.ans` bei, das einige Spezialfälle abdeckt. Wenn Sie mit **WRONG ANSWER** zu kämpfen haben, testen Sie, ob Ihr Programm mit diesen Fällen zurecht kommt, zum Beispiel, indem Sie diese Testfälle in `sample.in` und `sample.ans` hineinkopieren.

Aufgabe 3 (Maximale Teilsumme in $O(n)$)

Die größte erlaubte Eingabegröße beträgt wieder $n = 10^6$. Implementieren Sie den Ansatz **Ü1-A2-(d)** und reichen Sie Ihre Lösung für das Problem **p1-a3-maxsum-n** ein.

Hinweis: Folgende *Zusatzaufgabe* ist nicht verpflichtend.

Aufgabe 4 (Ausgabe der Indizes i und j).

Modifizieren Sie Ihre Lösung aus Aufgabe 2 oder 3, sodass diese nicht nur die größtmögliche Summe ausgibt, sondern Zeilen der Form (i, j, s) , wobei i und j diejenigen Indizes sind, die zur maximalen Summe $s = A[i] + \dots + A[j]$ führen (mit $i = 0$ und $j = -1$, falls die leere Summe $s = 0$ bestmöglich ist). Wenn es verschiedene Indexpaare (i, j) gibt, die alle zur gleichen maximalen Summe führen, dann bevorzugen Sie das lexikographisch kleinste Paar, also das Paar (i, j) , das für alle möglichen Lösungen (i', j') entweder $i < i'$ oder $i = i'$ und $j \leq j'$ erfüllt.

Die größte erlaubte Eingabegröße beträgt wieder $n = 10^6$. Reichen Sie Ihre Lösung für das Problem **p1-a4-maxsum-ij** ein.

Listing 3: `sample.in`

1	4	1	2	3	4			
2	4	2	-1	3	-1			
3	5	1	-1	2	3	-4		
4	7	2	-3	5	-1	-2	4	1

Listing 4: `sample.ans`

1	(0,3,10)
2	(0,2,4)
3	(0,3,5)
4	(2,6,7)

Hinweis zum Beispiel: Für die dritte Instanz mit maximaler Teilsumme 5 ist das Indexpaar $(0, 3)$ auszugeben, allerdings liefert auch $(2, 3)$ diese Summe.