



HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN · FH
MÜNCHEN

ACM Problem 111 – History Grading

von Stefan Gohlke

Ausgewählte Probleme aus dem ACM
Programming Contest

Wintersemester 09/10, Doina Logofătu

Inhalt

- 1) Problembeschreibung
- 2) Beispiel / Einordnung
- 3) Eingabe
- 4) Ausgabe
- 5) Lösung 1
- 6) Lösung 2



Problembeschreibung

- **Benotung einer Geschichtsklausur**
- **Ereignisse der Geschichte in korrekte Reihenfolge bringen**
- **Wenn alle Ereignisse korrekt geordnet wurden: volle Punktzahl**
- **Ansonsten 1 Punkt je Ereignis, das sich in der Reihenfolge von Ereignissen befindet, die richtig sortiert wurden.**
 - **Längste Reihenfolge**
 - **Muss nicht zusammenhängen**



Beispiel

- **Beispiel:**

- **Lösung:** 4 – 2 – 3 – 1
- **Schüler:** 4 – 1 – 2 – 3
- **Punkte:** 4 – 2 – 3 = 3 Punkte

- **Einordnung:**

- **Dynamische Programmierung**
- **Längste gemeinsame Teilfolge (LCS)**



Eingabe

- **n: Anzahl der Ereignisse (int; 2 bis 20)**
- **c: Musterlösung (Zeile von „n“ int; wie „n“)**
- **r: Antworten der Schüler (x Zeilen von „n“ int; wie „n“)**

- **Beispiel:**
 - **4 (Ereignisse)**
 - **4 2 3 1 (Musterlösung)**
 - **1 3 2 4 (Schülerantwort)**
 - **3 2 1 4 (Schülerantwort)**
 - **...**



Eingabe

- **Stolperfalle: Eingabe in chronologische Reihenfolge**

–	Position:	1	2	3	4	5	6	7	8	9	10
–	Eingabe:	3	1	2	4	9	5	10	6	8	7
–	Reihenfolge:	2	3	1	4	6	8	10	9	5	7

Aus

ergebnisse[i] = scanner.nextInt();

wird

ergebnisse[(scanner.nextInt()) - 1] = i;



Ausgabe

- **p: Punkte (je Schülerantwort ein int; 1 bis n)**
- **Beispiel:**
 - **1 (Punkte für Schülerantwort)**
 - **2 (Punkte für Schülerantwort)**
 - **...**



Lösung 1

- Lösung 1: eigener Lösungsansatz

Position	1	2	3	4	5	6	7	8	9	10
Musterlösung	2	3	1	4	6	8	10	9	5	7
Schülerantwort	1	2	3	4	5	6	7	8	9	10
Länge	1	1	2	3	4	4	5	5	6	6



Lösung 2

- Lösung 2: LCS-Algorithmus

	j	0	1	2	3	4	5	6	7	8	9
i	y_j		4	90	7	10	70	2	71	81	0
0	x_i	0	0	0	0	0	0	0	0	0	0
1	10	0	↑0	↑0	↑0	↘1	←1	←1	←1	←1	←1
2	4	0	←1	←1	←1	↑1	↑1	↑1	↑1	↑1	↑1
3	20	0	↑1	↑1	↑1	↑1	↑1	↑1	↑1	↑1	↑1
4	10	0	↑1	↑1	↑1	↘2	←2	←2	←2	←2	←2
5	40	0	↑1	↑1	↑1	↑2	↑2	↑2	↑2	↑2	↑2
6	2	0	↑1	↑1	↑1	↑2	↑2	↘3	←3	←3	←3
7	0	0	↑1	↑1	↑1	↑2	↑2	↑3	↑3	↑3	↘4
8	60	0	↑1	↑1	↑1	↑2	↑2	↑3	↑3	↑3	↑4

Quelle: Doina Logofătu, Grundlegende Algorithmen mit Java, Vieweg Verlag, ISBN 978-3-8348-0369-6, 2008, Seite 246



Vielen Dank für Ihre Aufmerksamkeit

