

## Junioraufgabe 2: Treffsicherheit

### J2.1 Lösungsidee

Betrachten wir zunächst einen konkreten Termin. Um herauszufinden, ob dieser Termin allseits beliebt ist, iterieren wir über alle Mitglieder der Gruppe. Für jedes Mitglied der Gruppe überprüfen wir, ob es einen anderen Termin gibt, den diese Person als besser bewertet hat. Wenn das der Fall ist, bezeichnen wir das als Konflikt. Der Termin ist dann nicht allseits beliebt.

Um einen Konflikt zu lösen, könnten wir entweder die Bewertung des gewählten Termins verbessern oder die Bewertung der anderen Termine verschlechtern. Wir müssen also für jede Person, die einen Konflikt auslöst, mindestens eine Änderung vornehmen. Eine Änderung pro Konflikt ist auch ausreichend, wenn wir immer die Bewertung des gewählten Termins auf grün setzen. Die Anzahl notwendiger Änderungen, um einen Termin allseits beliebt zu machen, ist also die Anzahl der Konflikte.

Um den besten Termin und die minimale Anzahl Änderungen zu finden, können wir also für alle Termine die Anzahl Konflikte berechnen und den Termin mit den wenigsten wählen.

### J2.2 Optimierung: Vorberechnung

Die Suche nach Konflikten für einen gewählten Termin kann so umgesetzt werden, dass für alle Personen alle Bewertungen betrachtet werden, um zu prüfen, ob eine davon besser als die Bewertung des gewählten Termins ist. Dies lässt sich jedoch optimieren. Es ist nämlich völlig ausreichend, zu wissen, was die beste Bewertung ist, die eine Person vergeben hat. Wenn die Bewertung für den gewählten Termin nicht dieser besten Bewertung entspricht, gibt es einen Konflikt.

Die beste Bewertung für jede Person können wir zu Beginn vorberechnen und in einem separaten Array speichern. Dieses nutzen wir dann für die folgende Suche nach Konflikten.

### J2.3 Laufzeit

Zur Umsetzung der Lösungsidee müssen wir für alle  $m$  Termine über alle  $n$  Personen iterieren, und für diese Person jeweils vergleichen, ob die Person einen der  $m - 1$  übrigen Terminvorschläge in der Präferenztable besser bewertet hat als den aktuell betrachteten. Bei der unoptimierten Variante iterieren wir dafür für jeden Termin und jede Person einmal über alle anderen Terminvorschläge, machen also ca.  $m \cdot n \cdot (m - 1)$  Schritte, was einer Laufzeit von  $O(m^2n)$  entspricht.

Bei der optimierten Version muss für jeden Termin und jede Person nur überprüft werden, ob die Bewertung der gespeicherten besten Bewertung dieser Person entspricht. Es werden also nur  $m \cdot n$  Schritte benötigt. Die Vorberechnung können wir durch einmaliges Iterieren über die gesamte Präferenztable umsetzen. Dies braucht auch nur  $m \cdot n$  Schritte. Damit liegt die Laufzeit insgesamt nur noch bei  $O(mn)$ .

In den gegebenen Beispieldateien ist  $n \leq 50$  und  $m \leq 80$ . Damit lässt sich auch die unoptimierte Variante locker in unter einer Sekunde auf allen Testfällen ausführen. Für die optimierte Variante wäre das auch z. B. noch für  $n = 1000$  und  $m = 1000$  möglich.

## J2.4 Lösung der Beispieltestfälle

Die gesuchten Werte sind in der folgenden Tabelle angegeben. Bei den Testfällen praefereenzen1.txt und praefereenzen5.txt gibt es je zwei verschiedene Termine, die gleich gut sind.

Testfall	minimale Anzahl Änderungen	beste Termine
praefereenzen0.txt	2	6
praefereenzen1.txt	1	2 oder 3
praefereenzen2.txt	0	4
praefereenzen3.txt	7	18
praefereenzen4.txt	14	22
praefereenzen5.txt	34	31 oder 56

Im Folgenden sind die kleinsten 4 Testfälle nochmal im Detail dargestellt. Dabei ist ein best-möglicher Termin grau unterlegt. Die grün unterlegten Tabelleneinträge müssen geändert werden, damit der Termin allseits beliebt wird.

praefereenzen0.txt						
?			?	?		
X	X	X	?	X	X	X
X	?	?	?	X	?	X
	?	X	X	?		
?	X	?	X		?	?

praefereenzen1.txt				
?	?	X	X	?
X	?	?	?	?
X	X		?	?
X			?	?

praefereenzen2.txt									
	?								
X	X	X		?	?	X			?
?	?			?	?	?			?
X	X	X	?	X	?	?	X	?	?
?	?	?	?	X	X	?	?	X	?
?	?	X		?	?	?	?	?	X
X	?	X		X	X	X	X		X
?	X	?		X	?	X	?	X	X

praefereenzen3.txt																			
		?	?		X	?	?	X	?	X	X	?	X		X	?	X	X	?
X	X	?		?	?	?		X	?		?	?	?	X			X	X	?
?	?		?	X	X	X	X	?	?	?	X	?	?	?	?	?		?	X
X	X	?	?	X			X			?	X	X	?	X	?		?	?	X
?	X	?	?	X	?	X	X	?	X	?	X	X	?	?	?	?		X	
	X	X	?	?	?	?	?		?	?	?	?	X	X	?	?	X	?	X
?	?	X	X	?	?	?	X		?		?	?	?	X	?	?			?
		X	?	X		?	X	X	?	?	X	?	?	X		?	X		?
?	?	?	?		?	?	?	?	?	?	?	?	X	?	?	?	X	X	X
X	?	?	?	X	?	?	?		X	?	X		?		?			X	
X	?	?	X	?	?		?	X	?	?	?	X	?	?		?		X	
	?		?	X	?	?	X	X	?			?	?	?	?	?		?	?
	?	X	?	?	?	?	X	?	X	?	X	X	?	?	?	X	?		X

## J2.5 Bewertungskriterien

Die Bewertungskriterien vom Bewertungsbogen werden hier erläutert (Punktabzug in []).

- [−1] **Lösungsverfahren fehlerhaft**  
Vorgesehen ist, dass bei jeder Änderung ein beliebiger Tabelleneintrag auf eine beliebige Bewertung gesetzt werden kann, also eine Änderung von „rot“ auf „grün“ nur einer Änderung entspricht. Wenn abweichende Annahmen getroffen werden, die die möglichen Änderungen einschränken, müssen diese zumindest in der Dokumentation explizit erwähnt werden.
- [−1] **Allseits beliebter Termin fehlerhaft umgesetzt**  
Ein allseits beliebter Termin sollte ein Termin sein, sodass für keine Person ein Termin existiert, welcher eine höhere Präferenz hätte. Dieser ist nicht notwendigerweise bei allen Personen „grün“ oder hat die wenigsten „roten“ Einträge. Davon auszugehen, dass der Termin mit der geringsten Summe zwangsläufig auch der beste Termin wird, ist nicht korrekt. Ist das Lösungsverfahren nur in diesem Punkt fehlerhaft, wird nur hier abgezogen und nicht zusätzlich bei Kriterium 4.
- [−1] **Ergebnisse schlecht nachvollziehbar**  
Zu einer vollständigen Ausgabe gehört mindestens die minimal nötige Anzahl Änderungen sowie die Angabe eines finalen Termins, der mit dieser Anzahl Änderungen allseits beliebt wird. Die Angabe der konkreten Änderungen ist nicht nötig. Wenn es gleichwertige „allseits beliebte Termine“ gibt, dann reicht es, wenn einer angegeben wird.
- [−1]  
**Beispiele fehlerhaft bzw. zu wenige oder ungeeignete Beispiele** Die Dokumentation soll Ergebnisse zu mindestens 3 der vorgegebenen Beispiele (praeferenzen1.txt bis praeferenzen5.txt) enthalten.