

### Aufgabe 1:

Wir möchten ermitteln ob ein Dungeon erkundet werden kann oder ob unsere Lebenskraft dafür nicht ausreicht. Unsere Lebenskraft wird über eine Anzahl an Herzen dargestellt und der Dungeon ist ein zweidimensionales, möglicherweise ausgefranztes Array.

Jeder Eintrag in dem zweidimensionalen Array stellt einen Raum dar. Der Eintrag wird beim betreten des Raumes auf die aktuelle Anzahl an Herzen addiert. Somit stellen negative Zahlen Gegner dar, die die Anzahl an Herzen verringern. Während der Erkundung muss unsere Anzahl an Herzen immer größer 0 sein.

Beim erkunden haben wir zwei Möglichkeiten:

- Wir steigen im aktuellen Raum  $i$  tiefer hinab und wechseln somit von Raum `dungeon[i][k]` in den Raum `dungeon[i][k+1]`.
- Wir steigen wieder hinauf und gehen weiter in den nächsten Raum in Richtung Ausgang. Dies würde ein Wechsel von `dungeon[i][k]` in den Raum `dungeon[i+1][0]` darstellen.

Wenn der Raum `dungeon[i+1][0]` nicht mehr existiert, verlassen wir den Dungeon und behalten die gammelten Herzen.

Bestimmen Sie die maximale Menge an Herzen mit denen wir einen beliebigen Dungeon verlassen können und geben Sie diese zurück. Sollte es keinen Weg geben bei dem die Anzahl an Herzen immer positiv bleibt, soll  $-1$  zurück gegeben werden.

Implementieren Sie dafür die Methode:

```
public static int erkunde(int [][] dungeon,
                          int raumNummer, int tiefe, int herzen)
```

Beispiel:

```
int [] keller0 = {2,-2,4,-5,2,1};
int [][] input = {keller0, {1,-5,6,-1,4}, {-10,1,-4,10}};
System.out.println(erkunde(input,0,0,2)); // ergibt 2
System.out.println(erkunde(input,0,0,1)); // ergibt -1
System.out.println(erkunde(input,0,0,4)); // ergibt 11
```

**Aufgabe 2:**

Problem: **Longest Common Subsequence (LCS):**

Wir betrachten ein Problem aus der Bio-Informatik: Gegeben seien zwei Ketten von Aminosäuren (**A**denin, **G**uanin, **C**ytosin und **T**hymine), die unterschiedliche lang sein dürfen.

Gesucht ist die längste gemeinsame Teilsequenz, der zwei Sequenzen  $A$  und  $B$ . Beachte den Unterschied zwischen Teilwort und Teilsequenz! Zur Bildung einer Teilsequenz schreitet man eine Sequenz von links nach rechts ab und kann für jedes Symbol entscheiden, ob es in die zu bildende Teilsequenz übernommen wird. Bei einem Teilwort wählt man lediglich den Start- und den End-Index.

Betrachten und verstehen Sie folgendes dynamisches Programm:

Mit  $L(i, j)$  bezeichnen wir die Länge der längsten gemeinsamen Teilsequenz der Sequenzen (Präfixe)  $a_0, \dots, a_i$  und  $b_0, \dots, b_j$ . Das Ergebnis lässt sich dann an der Stelle  $L(n, m)$  auslesen.

Zu beachten sind die Randfälle:

$$L(i, j) = 0 \text{ falls } i < 0 \vee j < 0 \vee i > n \vee j > m .$$

Der Zusammenhang zwischen den Teillösungen ergibt sich durch folgende Formel:

$$L(i, j) = \begin{cases} 1 + L(i - 1, j - 1) & \text{wenn } a_i = b_j \\ \max(L(i - 1, j), L(i, j - 1)) & \text{sonst} \end{cases}$$

1. Wenden Sie den Algorithmus auf die unten stehende leere Tabelle an. [Oben Links ist  $L(0, 0)$ ]
2. Geben Sie die Laufzeit des Algorithmus an.

Beispiele:

	<u>G</u>	<u>A</u>	<u>T</u>	<u>T</u>	<u>A</u>	<u>C</u>	<u>A</u>
<u>G</u>	1	1	1	1	1	1	1
<u>T</u>	1	1	2	2	2	2	2
<u>T</u>	1	1	2	3	3	3	3
<u>A</u>	1	2	2	3	4	4	4
<u>C</u>	1	2	2	3	4	5	5
<u>A</u>	1	2	2	3	4	5	6

	<u>A</u>	<u>G</u>	<u>C</u>	<u>T</u>	<u>A</u>	<u>G</u>	<u>C</u>
<u>G</u>	0	1	1	1	1	1	1
<u>A</u>	1	1	1	1	2	2	2
<u>C</u>	1	1	2	2	2	2	3
<u>G</u>	1	2	2	2	2	3	3
<u>T</u>	1	2	2	3	3	3	3
<u>A</u>	1	2	2	3	4	4	4

	<u>A</u>	<u>A</u>	<u>G</u>	<u>A</u>	<u>C</u>	<u>T</u>	<u>A</u>	<u>A</u>	<u>G</u>	<u>C</u>	<u>A</u>
<u>G</u>											
<u>A</u>											
<u>C</u>											
<u>G</u>											
<u>T</u>											
<u>A</u>											
<u>C</u>											