

Problem

Gegeben sei eine $m \times n$ Matrix, deren Einträge den Kosten entsprechen, die man beim Überschreiten eines Feldes bezahlen muss. Ziel ist es, einen optimalen Weg von der Zelle links oben $(0, 0)$ zur Zelle rechts unten $(m - 1, n - 1)$ zu finden, auf dem man möglichst wenig Wegzoll bezahlen muss. Von einer Zelle aus darf man nur nach rechts oder nach unten gehen.

Beispiel

```
{ 4 7 8 6 4 }
{ 6 7 3 9 2 }
{ 3 8 1 2 4 }
{ 7 1 7 3 7 }
{ 2 9 8 9 3 }
```

```
{ 4 7 8 6 4 }
{ 6-7-3, 9 2 }
{ 3 8 1-2 4 }
{ 7 1 7 3-7 }
{ 2 9 8 9 3 }
```

Rekursion

Kosten um Zelle (m, n) zu erreichen =
Kosten auf Zelle (m, n) +
Minimum (Kosten um Zelle $(m - 1, n)$ zu erreichen,
Kosten um Zelle $(m, n - 1)$ zu erreichen)

Codefragment

```
int[][] cost =
{
    { 4, 7, 8, 6, 4 },
    { 6, 7, 3, 9, 2 },
    { 3, 8, 1, 2, 4 },
    { 7, 1, 7, 3, 7 },
    { 2, 9, 8, 9, 3 }
};

void setup() {
    noLoop();
}

void draw() {
    println (minCost(4, 4));
}
```

```
int minCost(int m, int n) {
    // wegkosten zur zelle (0,0)

    // boundaries beachten!

    // rekursion
}
```