

BQN – eine moderne Array-Programmiersprache

Leah Neukirchen <leah@vuxu.org>

Aktiv Symposium München '22
4. Juni 2022

Was ist Array-Programmierung?

- Existiert seit den 60ern: Ursprung in Ken Iverson's APL
- Direkte Operationen auf Arrays als Datentypen
- "No stinking loops"
- heutzutage: funktionaler Stil, Funktionen höherer Ordnung
- *tacit programming*: Programmierung ohne benannte Variablen

Die BQN-Tastatur

~ ~ \ ~	! ◉ 1 ~	@ ◉ 2 ¨	# ◉ 3 =	\$ ◉ 4 ⸮	% ◉ 5 ´	^ ◉ 6 ~	& ◉ 7	* ◉ 8 ∞	(<) > 9 - 0 •	_ √ - ÷	+ * = ×	Backspace
Tab	Q ✓ q φ	W W w w	E ∈ e ∈	R r r ↑	T † t ^	Y y	U u u	I ∈ i ∈	O ∃ o ∃	P ∟ p π	{ † } † [†] †	 \
Caps Lock	A ↖ a ↘	S S s s	D d ↓	F F f f	G G g g	H « h ~	J j °	K ◉ k ◉	L » l ~	: · ; ◊	" · ' ↵	Enter
Shift	Z ↯ z ↵	X X x x	C c ↓	V ↓ v v	B ⊎ b ⊎	N n	M ≠ m ≡	< ≤ , ∞	> ≥ . =	? † / ≠	Shift	

Rechnen mit Vektoren

$$23 + 19$$

42

$$22 + 1_2_3$$

$\langle 23 \ 24 \ 25 \rangle$

$$23_24_25 + 19_20_21$$

$\langle 42 \ 44 \ 46 \rangle$

Rechnen mit Vektoren

23_24_25 +^r 19_20_21

┌
42 43 44
43 44 45
44 45 46
└

Berechnung des Durchschnitt

$$a \leftarrow 5_6_7_8$$

< 5 6 7 8 >

$$5+6+7+8$$

26

$$+ 'a$$

26

$$\neq a$$

4

Berechnung des Durchschnitts

$$(+ ' a) \div \neq a$$

6.5

$$(+ ' \div \neq) a$$

6.5

$$\text{Avg} \leftarrow + ' \div \neq$$

$$\text{Avg } a$$

6.5

Advent of Code 2021: Day 9: Smoke Basin

Smoke flows to the lowest point of the area it's in. For example, consider the following heightmap:

```
2199943210
3987894921
9856789892
8767896789
9899965678
```

Your first goal is to **find the low points**—the locations that are lower than any of its adjacent locations.

The **risk level** of a low point is 1 plus its height. **What is the sum of the risk levels of all low points on your heightmap?**

Laden einer Datei

```
d ← > '0' -~ •FLines"day09"
```

```
┌
2 1 9 9 9 4 3 2 1 0
3 9 8 7 8 9 4 9 2 1
9 8 5 6 7 8 9 8 9 2
8 7 6 7 8 9 6 7 8 9
9 8 9 9 9 6 5 6 7 8
└
```

Zeilen-Shift

$\infty \gg \sim d$

∞	2	1	9	9	9	4	3	2	1
∞	3	9	8	7	8	9	4	9	2
∞	9	8	5	6	7	8	9	8	9
∞	8	7	6	7	8	9	6	7	8
∞	9	8	9	9	9	6	5	6	7

Drehung einer Matrix

$\otimes \phi d$

9	8	9	3	2
8	7	8	9	1
9	6	5	8	9
9	7	6	7	9
9	8	7	8	9
6	9	8	9	4
5	6	9	4	3
6	7	8	9	2
7	8	9	2	1
8	9	2	1	0

Iterierte Drehung

$$(\mathbb{Q} \circ \phi \otimes (\uparrow 4)) \text{ d}$$

2	1	9	9	9	4	3	2	1	0
3	9	8	7	8	9	4	9	2	1
9	8	5	6	7	8	9	8	9	2
8	7	6	7	8	9	6	7	8	9
9	8	9	9	9	6	5	6	7	8
9	8	9	3	2					
8	7	8	9	1					
9	6	5	8	9					
9	7	6	7	9					
9	8	7	8	9					
6	9	8	9	4					
5	6	9	4	3					
6	7	8	9	2					
7	8	9	2	1					
8	9	2	1	0					
8	7	6	5	6	9	9	9	8	9
9	8	7	6	9	8	7	6	7	8
2	9	8	9	8	7	6	5	8	9
1	2	9	4	9	8	7	8	9	3
0	1	2	3	4	9	9	9	1	2
0	1	2	9	8					
1	2	9	8	7					
2	9	8	7	6					
3	4	9	6	5					
4	9	8	9	6					
9	8	7	8	9					
9	7	6	7	9					
9	8	5	6	9					
1	9	8	7	8					
2	3	9	8	9					

Kombination von Drehung *unter* Zeilenshift

$$\{\infty \rightarrow \gg \sim \odot (\otimes \circ \phi \otimes \otimes \times) d\} \updownarrow 4$$

∞	2	1	9	9	9	4	3	2	1
∞	3	9	8	7	8	9	4	9	2
∞	9	8	5	6	7	8	9	8	9
∞	8	7	6	7	8	9	6	7	8
∞	9	8	9	9	9	6	5	6	7

3	9	8	7	8	9	4	9	2	1
9	8	5	6	7	8	9	8	9	2
8	7	6	7	8	9	6	7	8	9
9	8	9	9	9	6	5	6	7	8
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

1	9	9	9	4	3	2	1	0	∞
9	8	7	8	9	4	9	2	1	∞
8	5	6	7	8	9	8	9	2	∞
7	6	7	8	9	6	7	8	9	∞
8	9	9	9	6	5	6	7	8	∞

∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	1	9	9	9	4	3	2	1	0
3	9	8	7	8	9	4	9	2	1
9	8	5	6	7	8	9	8	9	2
9	8	7	6	7	8	9	6	7	8

$L' \{ \infty \rightarrow \gg \sim \odot (\otimes \circ \phi \otimes \times) d \} \updownarrow 4$

1	2	1	7	4	3	2	1	0	1
2	1	5	6	7	4	3	2	1	0
3	5	6	5	6	7	4	7	2	1
7	6	5	6	7	6	5	6	7	2
8	7	6	7	6	5	6	5	6	7

$$1 \leftarrow d < \lfloor \{ \infty \rightarrow \} \circ (\phi \otimes x) d \rfloor \updownarrow 4$$

0	1	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0

$(1+d) \times 1$

┌
0 2 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0
0 0 6 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 6 0 0 0

└

$+ \text{'} \leftrightarrow (1+d) \times 1$

15

Warum?

- Eleganz
- Neue Denkmuster und Denkweisen
- Hohe Informationsdichte, extrem konzise Programme
- Intellektuelle Kurosität und Spielerei, Code-Golfing
- Gut geeignet für numerische Probleme, Prototypen, als Alternative zu Numpy, MATLAB, etc.
- Open Source (ISC), gute und recht schnelle Implementierung (CBQN), ausführliche Doku, nette und hilfreiche Community

Mehr Information

- <https://mlochbaum.github.io/BQN>
- Teil II: leahneukirchen.org/blog/archive/2021/12/surveying-lava-basins-with-bqn-and-fixpoints.html
- Day 6: leahneukirchen.org/blog/archive/2021/12/counting-lanternfish-with-bqn-and-linear-algebra.html
- github.com/leahneukirchen/adventofcode2021