

Lagebesprechung

Woche 3: Dienstag

Holger Dell

Planung für Heute

1. Kurze Wiederholung der Folien zur Analyse von Algorithmen
2. Meta-Kognition über das Lernen
3. 8:15-9:15 Gruppenarbeit & Helpdesks
4. 9:15 Lösung der ★-Aufgabe 2D-Hügel, Teil c)
5. ca. 9:30 Lösung der Aufgaben von heute.

Lernziele von ALGO-1

Ich will Ihnen folgende **Fähigkeiten** vermitteln:

- wichtige Algorithmen und Datenstrukturen **kennen und benutzen.**
- Algorithmen auf Ihre Korrektheit und Laufzeit **analysieren.**
- algorithmische Ideen selbst **entwickeln.**
- eine algorithmische Idee zu Pseudocode und Code **konkretisieren.**
- schriftlich und mündlich über Algorithmen **argumentieren.**

Die Klausur evaluiert, inwiefern diese Lernziele erreicht wurden.

Angebote dieser Vorlesung

- **Vorlesungsvideos.** Sie lernen wichtige Algorithmen, Datenstrukturen und Methoden kennen.
- **Übungsaufgaben / Gruppenarbeit.** Sie trainieren Fähigkeiten, um die Lernziele zu erreichen.
- **Buch.** Sie lernen die korrekte und schriftliche Fachsprache kennen.
- **Helpdesks / Lernzentrum-Discord. Wir sind für Sie da!** Wenn Sie
 - etwas nicht verstehen oder eine Frage haben
 - nicht wissen, was Sie tun sollen
 - sich mit den Inhalten alleine fühlen
 - nicht sicher sind, ob Ihr Lösungsvorschlag korrekt ist

Exkurs: Kognitiver Konstruktivismus (nach Jean Piaget)

- Wissen und Fähigkeiten werden im Gehirn **konstruiert**.
- **Schema**. Besteht aus drei Teilen:
 - a. Wahrgenommene oder erkannte Situation.
 - b. Mit der Situation verknüpfte körperliche oder mentale Handlung.
 - c. Erwartungshaltung dazu, welchen Effekt die Handlung hat.
- Gehirn **korrigiert** ein Schema, wenn bei einer Aktivität ein “Fehler” auftritt, also wenn die Erwartungshaltung nicht zum tatsächlich eingetretenen Effekt passt:
 - a. Wahrnehmung der Situation wird verschärft
 - b. Handlung wird zielgerichteter
 - c. Erwartungshaltung wird korrigiert
- Kognitiver Konstruktivismus \Rightarrow Lernen braucht **Aktivität** und **Fehler**.

Beispiel

$$T(n) = T(n/3) + cn$$

In 1. Rekursionsstufe: $cn/3$

In 2. Rekursionsstufe: $cn/9$

...

In k . Rekursionsstufe: $cn/3^k$

Fertig wenn $k = \log_3 n$.

Daher ist $T(n) = cn/3^k = c$.

$$T(n) = T(n/3) + cn$$

$$= T(n/9) + cn/3 + cn$$

$$= T(n/81) + cn/9 + cn/3 + cn$$

= ...

$$= cn * (1 + 1/3 + 1/3^2 + 1/3^3 + \dots)$$

$$\leq cn * 2$$

Übungsaufgaben als **emotionaler Prozess**

Beobachtungen.

- Stecken bleiben, Fehler machen, nicht sicher ob korrekt

Mögliche Emotionen.

- Frustration, Wut, Angst, Neid, Minderwertigkeitsgefühle
- Neugierde, Ehrgeiz

Diese Prozesse sind -in Maßen- **notwendige Bedingung** zum Lernen.

Musterlösungen als Emotionsregulation

Emotionale Funktion.

- Sie erhalten ein Gefühl von Kontrolle.
- Sie fühlen sich weniger frustriert.
- Sie fühlen sich, als würden Sie die Lösung verstehen.

Problem.

- Falsches Gefühl der Sicherheit.
- Wenn man es wirklich verstanden hat, braucht man keine Musterlösung.
- Fehlende Selbstkorrektur und fehlendes soziales Feedback

Sinn der Übungen

Normale Aufgaben.

- kleinteilige Trainingsziele
- einfache und mittelschwere Aufgaben:
wichtige Algorithmen, Datenstrukturen und Methoden **verstehen** und **benutzen**.
- schwere Aufgaben:
neue Algorithmen und Datenstrukturen **entwickeln**.

★-Aufgaben.

- wie normale Aufgaben
- Gedanken und Ideen in korrekter Fachsprache verständlich **aufschreiben**.

Mission dieser Veranstaltung

- Zahl der **Kontakte zwischen Lehrpersonal und Studis** so groß wie möglich
 - **Individuelle Betreuung** und Beantwortung Ihrer Fragen
 - Fokus auf diejenigen Studis, die **jetzt** aktiv am Kurs teilnehmen
-
- **Wir wollen Ihren Lernerfolg!**
 - Sie haben die Kapazität, es zu schaffen.

Planung für Heute

1. Kurze Wiederholung der Folien zur Analyse von Algorithmen
2. Meta-Kognition über das Lernen
3. 8:15-9:15 Gruppenarbeit & Helpdesks
4. 9:15 Lösung der ★-Aufgabe 2D-Hügel, Teil c)
5. ca. 9:30 Lösung der Aufgaben von heute.

Erste -Aufgabe (2D-Hügel)

- **176 Abgaben.**
- bewertet innerhalb von 1,5 Tagen.
- ca. **2/3** wurden “akzeptiert”.
- individuelle Fragen zur Korrektur direkt Freitags im Helpdesk geklärt.

Lösung ★-Aufgabe c)

- Zunächst eine **Beobachtung**. Jede $(n \times n)$ -Matrix M hat einen Hügel.
- **Beweis**.
 - M hat einen maximalen Eintrag, z.B. $M[i][j]$.
 - Dann ist $M[i][j] \geq$ Nachbarn, und daher ist $M[i][j]$ auch ein Hügel.

Matrizen

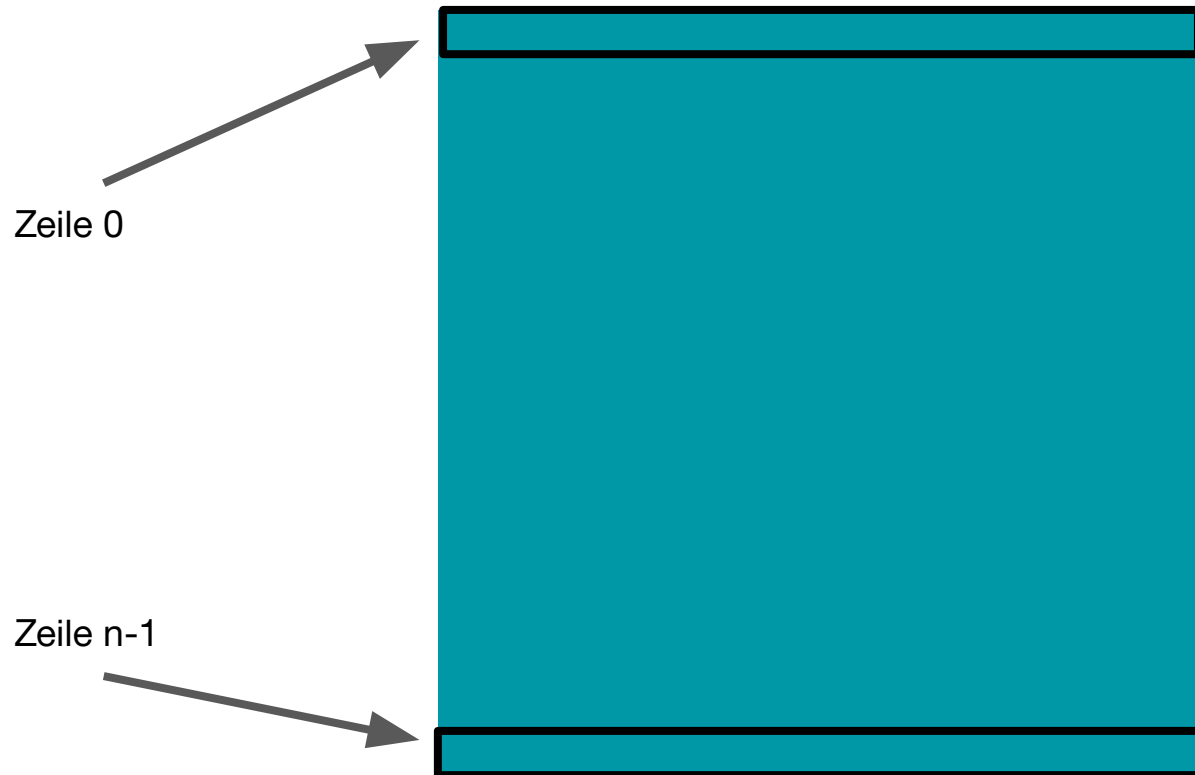
$M[i][j]$

Zeilenindex

Spaltenindex

Reihenindex

Matrizen

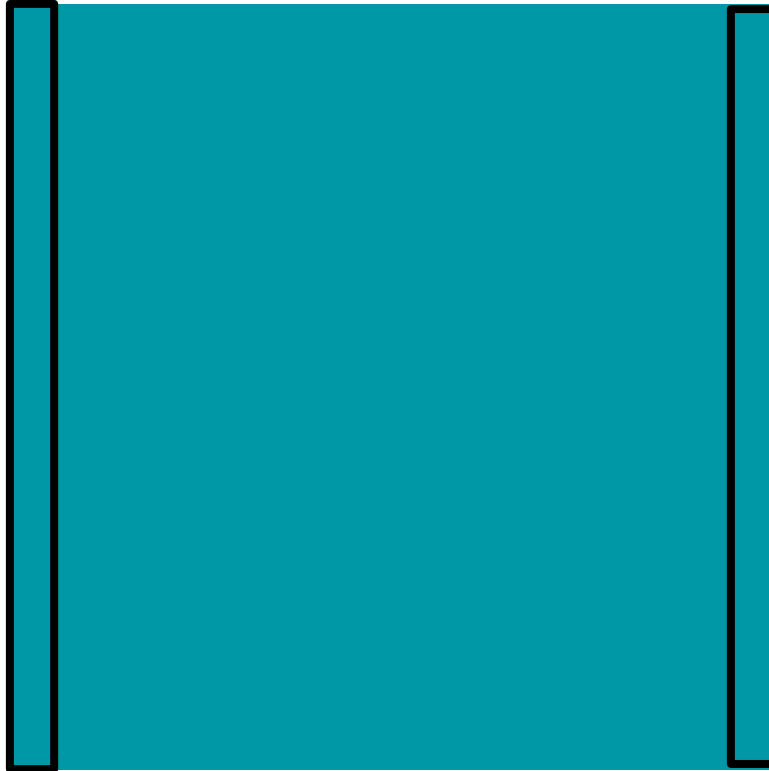


Matrizen

Spalte 0



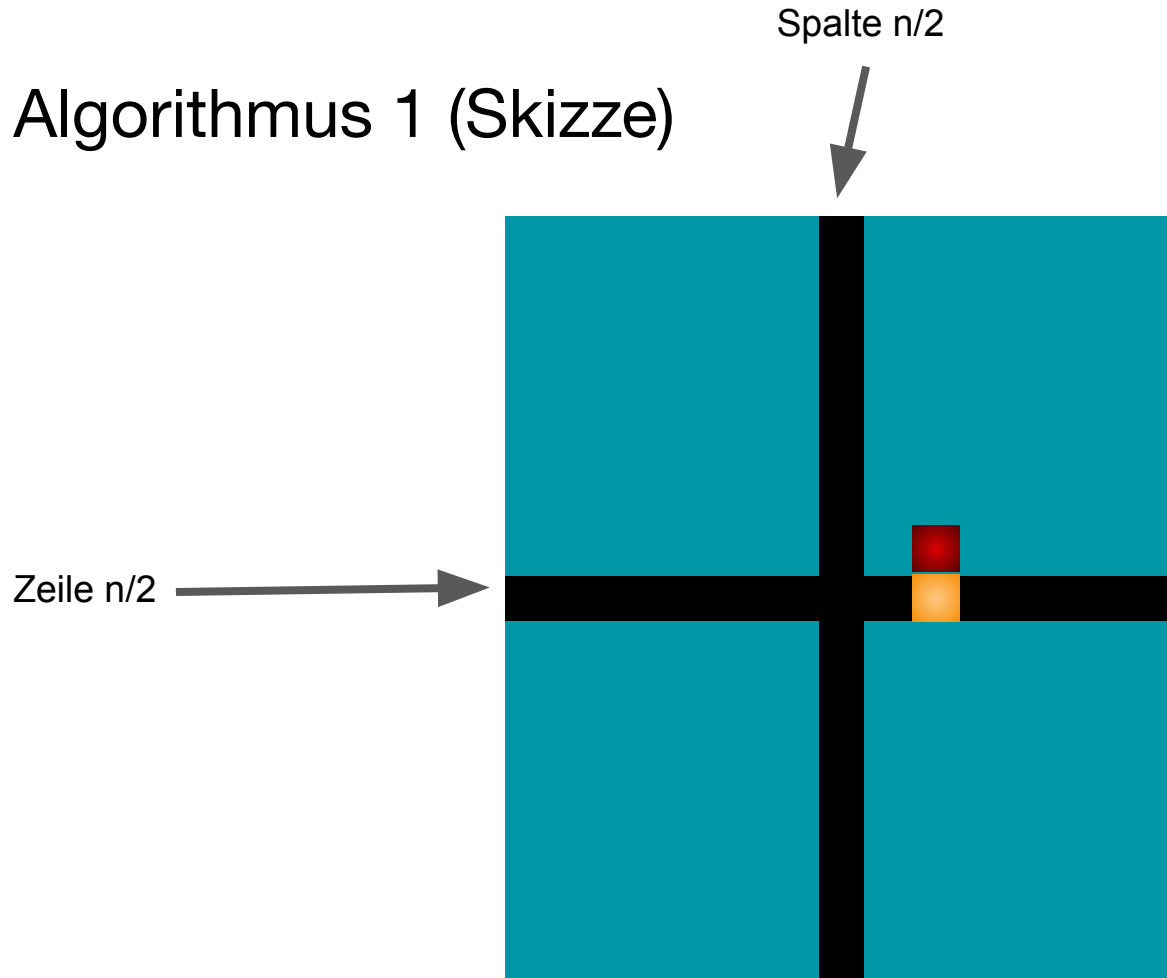
Spalte n-1





Intuition


“Immer den Berg hochgehen führt zu einem Hügel.”

Algorithmus 1 (Skizze)

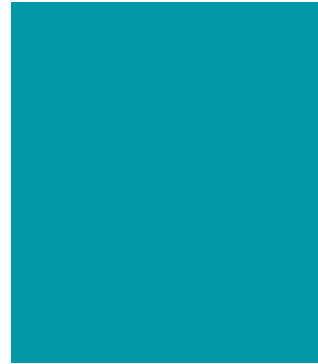






1. Finde Maximum  unter den Einträgen im Mittelkreuz.

2. Finde Nachbar  echt größer als .

3. Gehe mit dem Quadranten, der  enthält, in Rekursion.

Algorithmus 1 (Skizze)



1. Finde Maximum  unter den Einträgen im Mittelkreuz.
2. Finde Nachbar  echt größer als .
3. Gehe mit dem Quadranten, der  enthält, in Rekursion.

Dieser Algorithmus hat Laufzeit $O(n)$,
aber ist nicht korrekt.

Algorithmus 1 (Laufzeit)

- Maximum auf Mittelkreuz finden braucht **cn** Zeit.
- rekursiver Aufruf reduziert **n** (=Zeilenzahl=Spaltenzahl) auf **n/2**.

$$T(n) = T(n/2) + cn = .. = cn + cn/2 + cn/4 + .. = 2cn = O(n)$$

Algorithmus 1 (Gegenbeispiel)

9	9	9	8	2	1	0
9	9	9	8	0	0	0
9	9	9	8	0	0	10
8	8	8	8	8	8	9
9	9	9	8	9	9	9
9	9	9	8	9	9	9
9	9	9	8	9	9	9

Algorithmus 1 (Gegenbeispiel)

Das ist kein Hügel

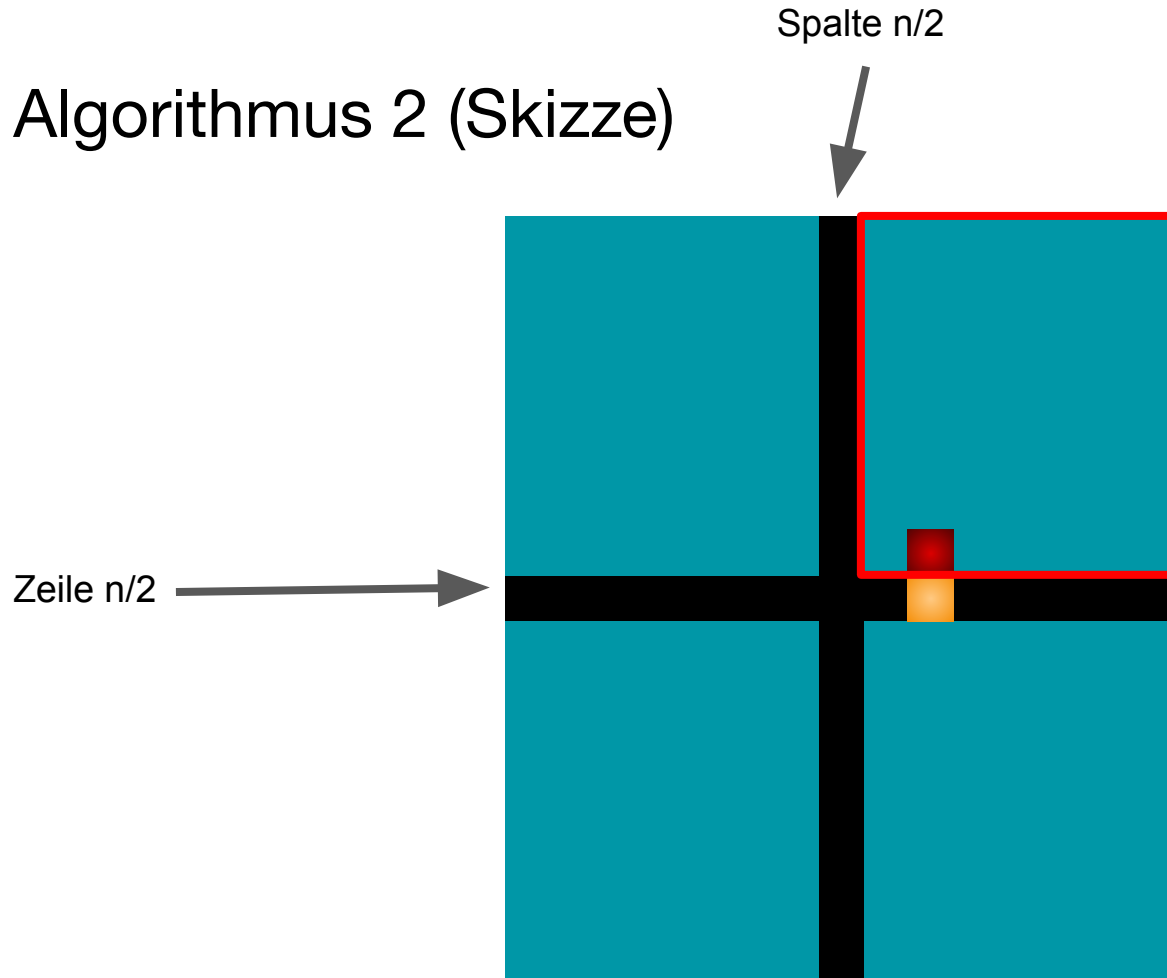


The image shows a 7x7 grid of numbers. The grid is as follows:



9	9	9	8	2	1	0
9	9	9	8	0	0	0
9	9	9	8	0	0	10
8	8	8	8	8	8	9
9	9	9	8	9	9	9
9	9	9	8	9	9	9
9	9	9	8	9	9	9


An arrow points from the text "Das ist kein Hügel" to the cell containing the number 2 in the first row, fifth column.

Algorithmus 2 (Skizze)

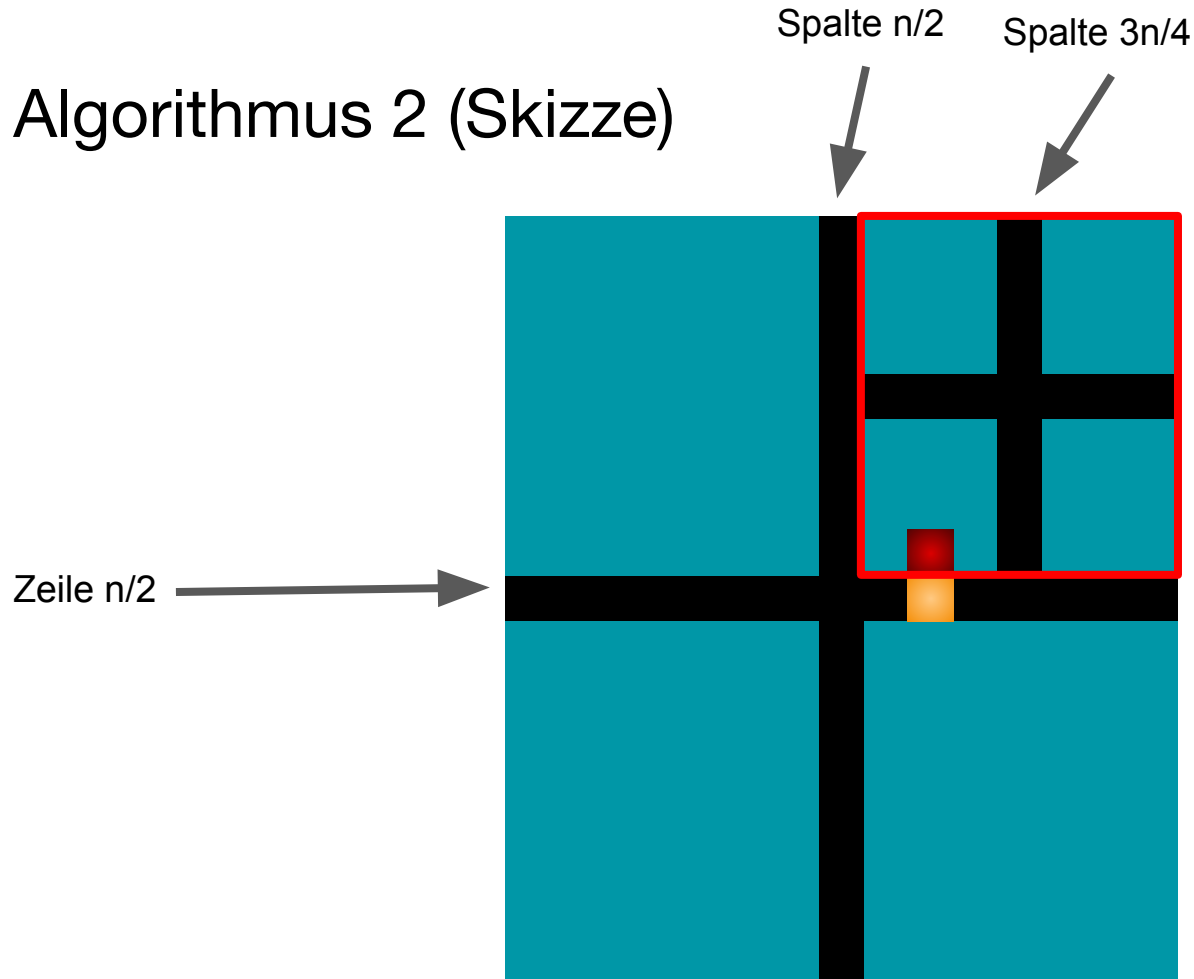



1. Finde Maximum  unter allen schwarzen Einträgen



2. Finde Nachbar  echt größer als .


3. Gehe mit dem Quadranten, der  enthält, in Rekursion.

Algorithmus 2 (Skizze)

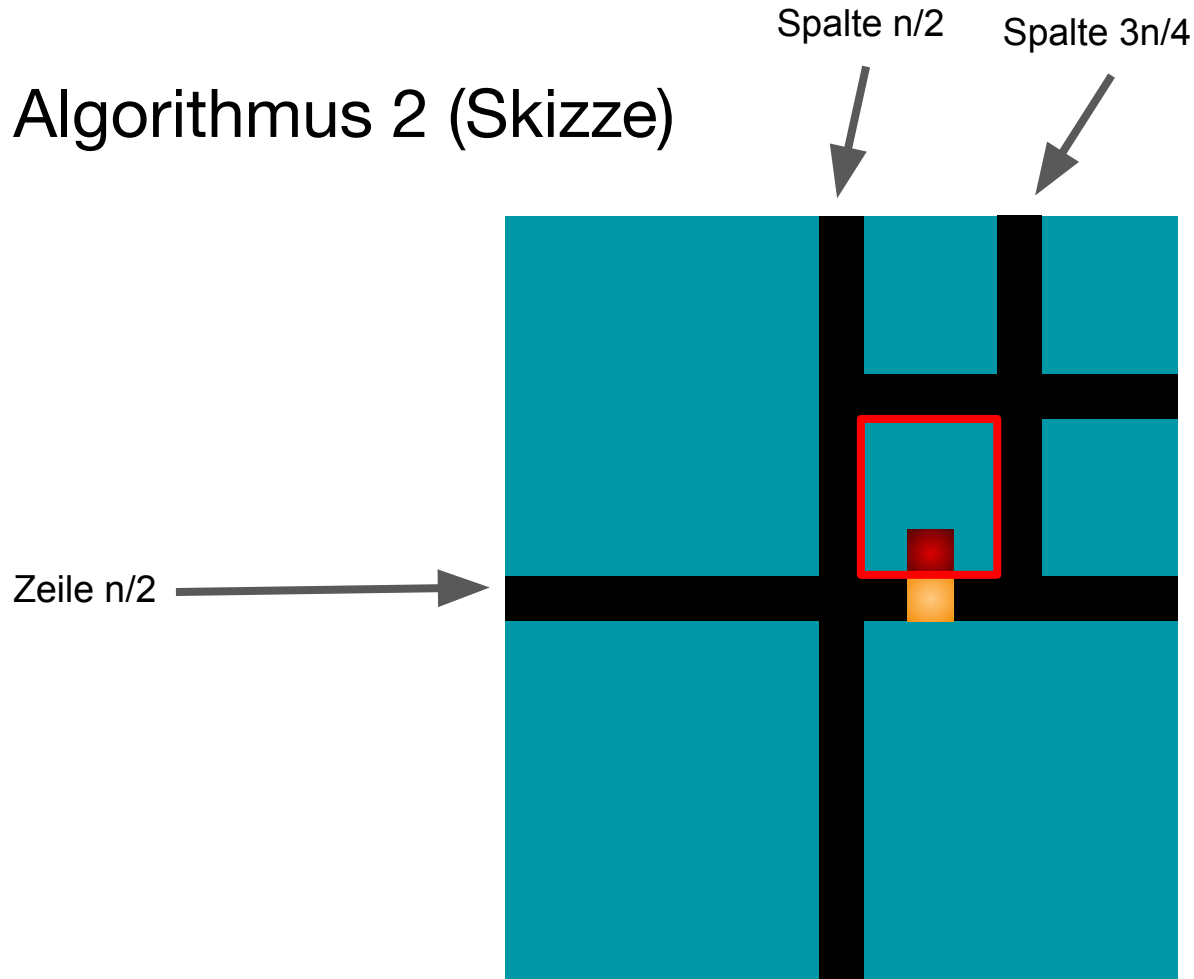



1. Finde Maximum  unter allen schwarzen Einträgen



2. Finde Nachbar  echt größer als .


3. Gehe mit dem Quadranten, der  enthält, in Rekursion.

Algorithmus 2 (Skizze)

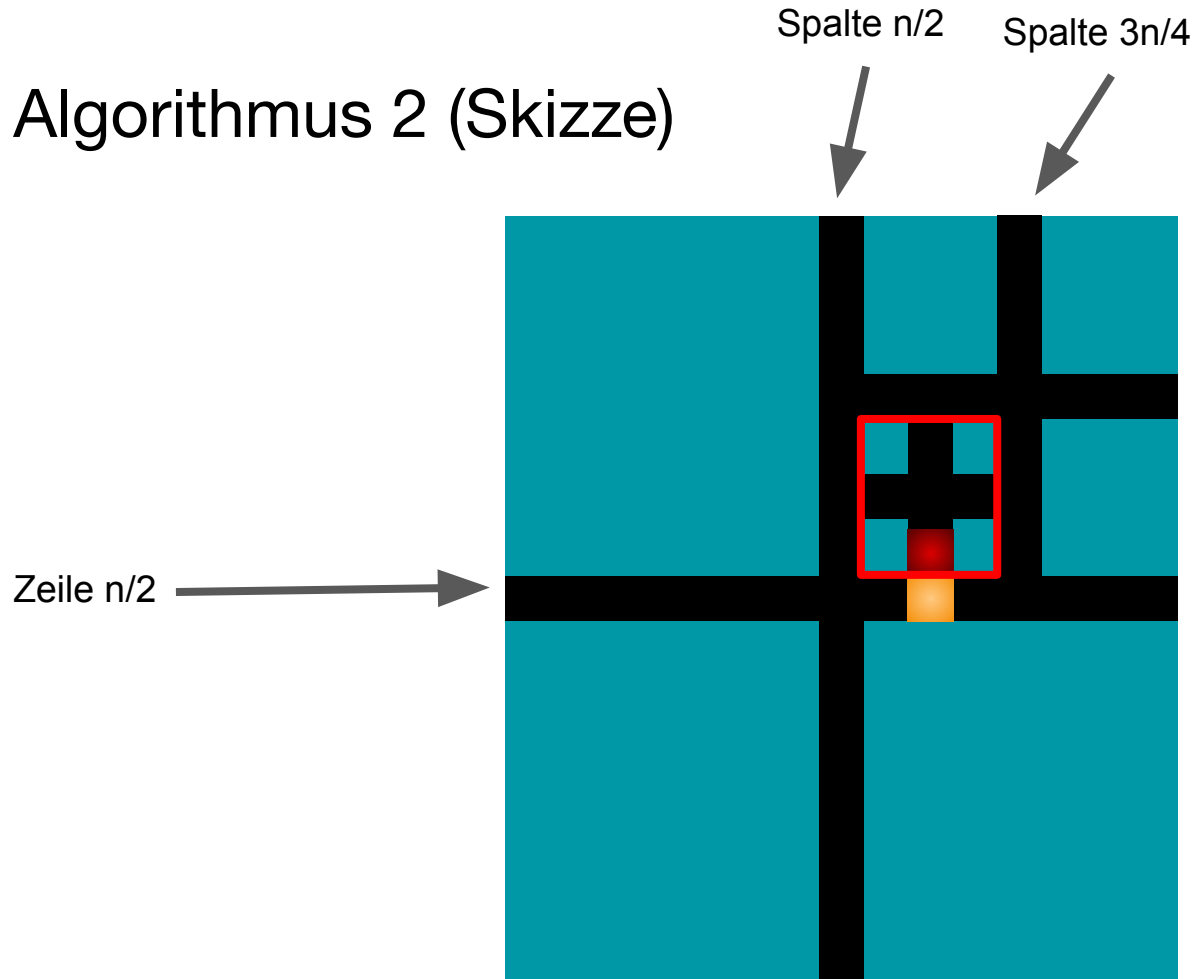



1. Finde Maximum  unter allen schwarzen Einträgen



2. Finde Nachbar  echt größer als .


3. Gehe mit dem Quadranten, der  enthält, in Rekursion.

Algorithmus 2 (Skizze)

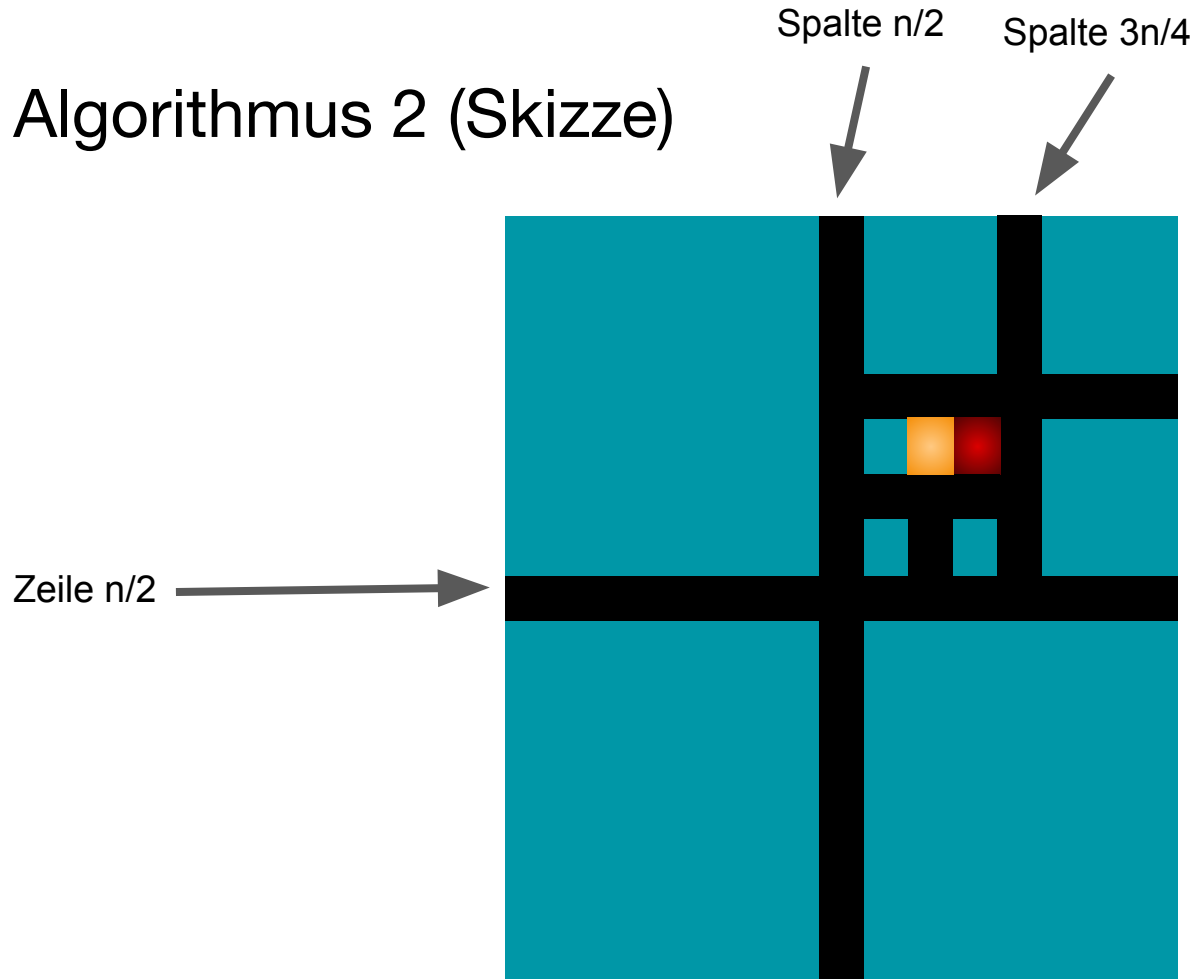



1. Finde Maximum  unter allen schwarzen Einträgen



2. Finde Nachbar  echt größer als .


3. Gehe mit dem Quadranten, der  enthält, in Rekursion.

Algorithmus 2 (Skizze)

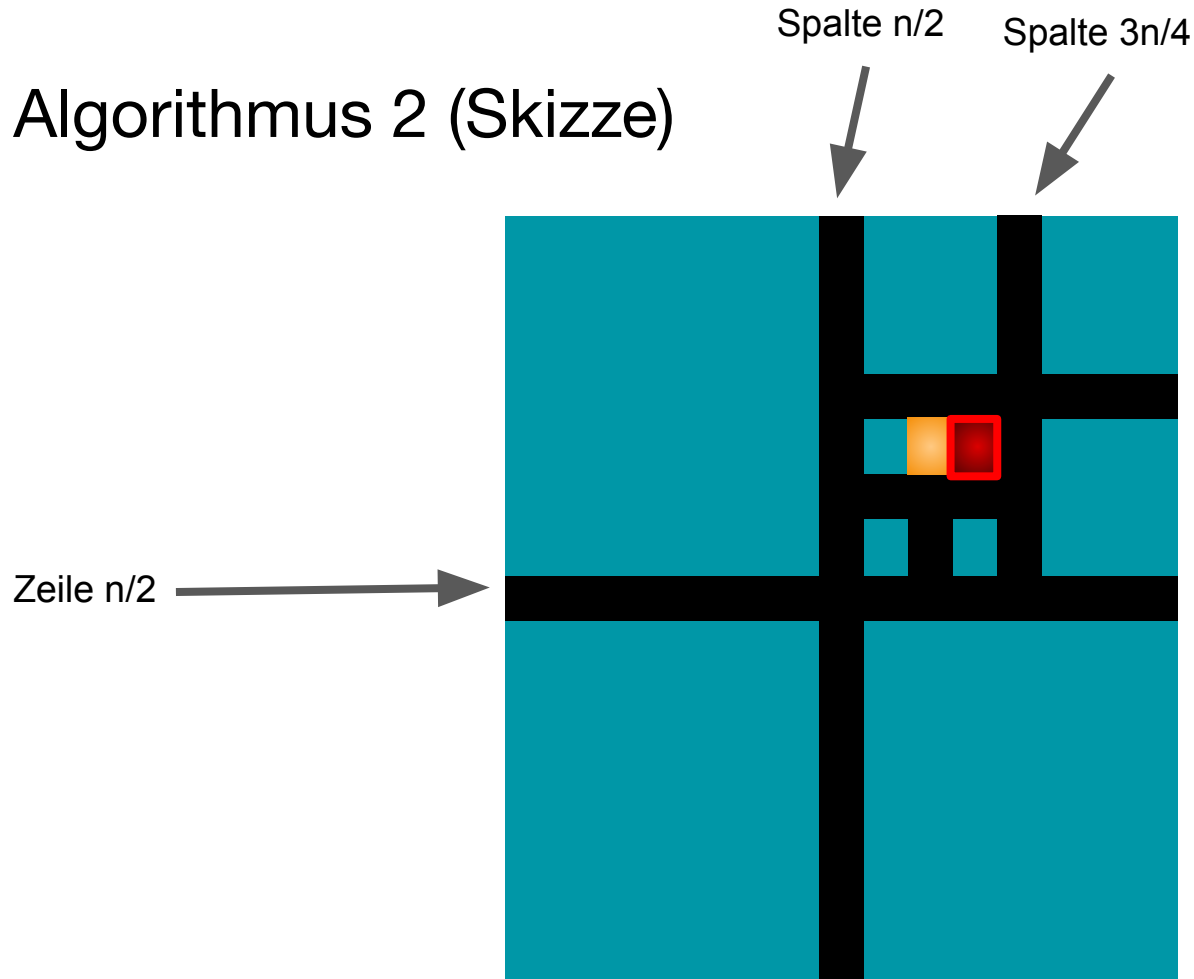



1. Finde Maximum  unter allen schwarzen Einträgen



2. Finde Nachbar  echt größer als .


3. Gehe mit dem Quadranten, der  enthält, in Rekursion.

Algorithmus 2 (Skizze)



1. Finde Maximum  unter allen schwarzen Einträgen

2. Finde Nachbar  echt größer als .

3. Gehe mit dem Quadranten, der  enthält, in Rekursion.

Algorithmus 2 (Korrektheit)

Zu Zeigen: “Der Algorithmus gibt einen Hügel aus.”

Beweis durch Widerspruch:

- Angenommen, die Ausgabe (i,j) von Algorithmus 2 ist kein Hügel.
- Dann ist ein Nachbar von (i,j) größer.
- Dieser Nachbar kann nicht schwarz markiert sein, denn (i,j) ist der maximale schwarze Eintrag.
- Daher wäre Algorithmus 2 in Rekursion gegangen und hätte nicht (i,j) ausgegeben. \downarrow