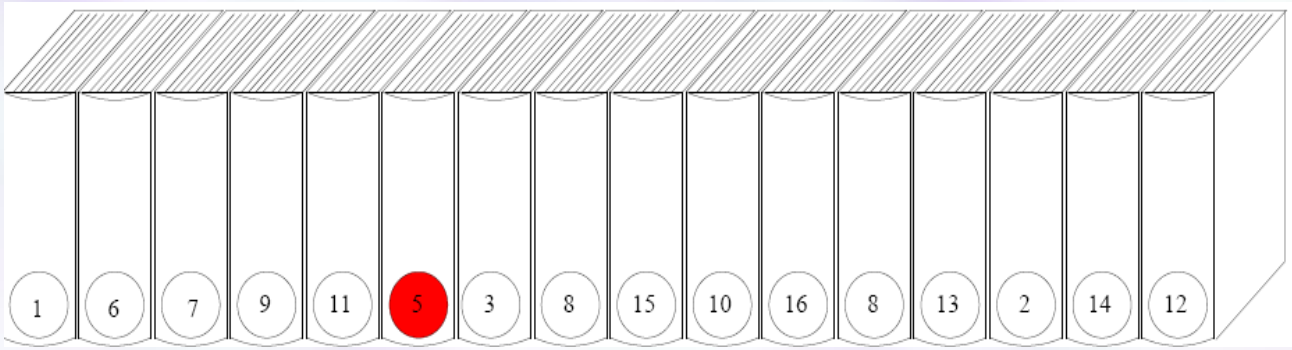
Sortieren durch Einfügen

Schon wieder aufräumen

- Schon wieder Aufräumen,
  dabei habe ich doch erst neulich...
- man findet alles schneller wieder
- Bücher auf Regal nach Titeln sortieren



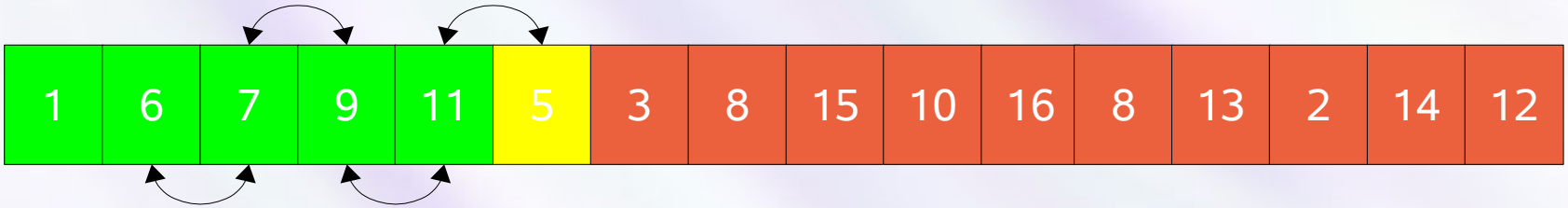
Wie schafft man Ordnung?

- Wie schafft man am schnellsten Ordnung?
- Natürliche Idee
  - bringe alle Bücher von links nach rechts im Regal fortschreitend an richtige Position

1. Das erste Buch bleibt zunächst an 1. Stelle
2. das zweite Buch wird nur mit erstem vertauscht, wenn es falsch steht
3. drittes Buch wird nur mit zweitem vertauscht, wenn es falsch steht, ggf. auch mit erstem
4. usw.

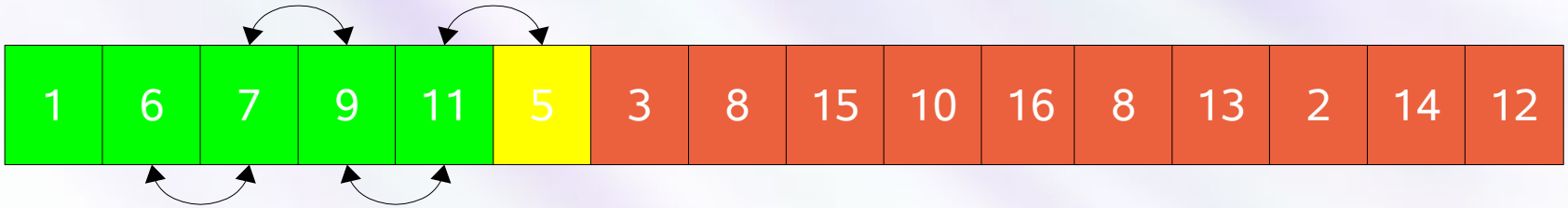
Sortieren des Bücherregals

- Allgemein nehmen wir an
  - alle Bücher links im Bücherregal bereits sortiert
  - wir eines hinzu und bringen es durch Vertauschen mit linken Nachbarn an richtige Position
- Statt der Buchtitel schreiben wir im folgenden Nummern, weil der Algorithmus dann einfacher zu erklären ist.



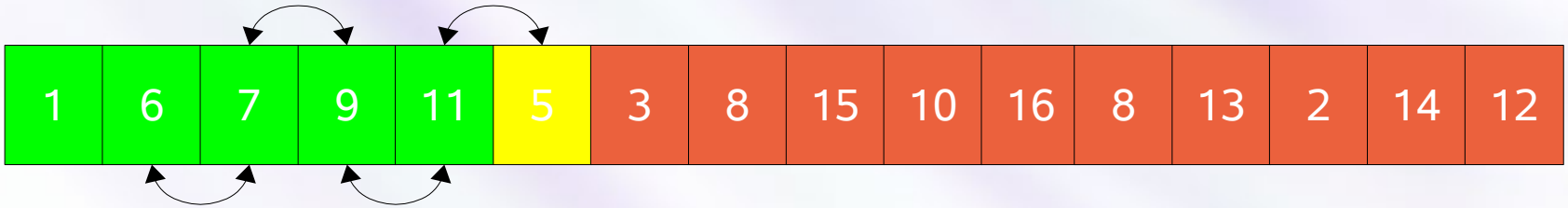
Sortieren des Bücherregals

- Beispiel
  1. linke fünf Bücher bereits sortiert
  2. Buch mit Nummer 5 hinzunehmen
     es steht offensichtlich falsch
  3. Vertausche zunächst mit Buch Nummer 11
  4. Vertausche mit Buch Nummer 9
  5. usw.



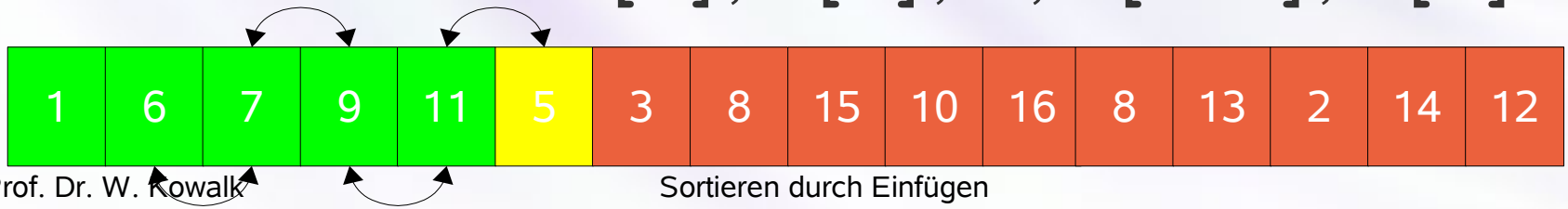
Sortieren des Bücherregals

- Beispiel (Fortsetzung)
  6. höre auf, wenn es einmal rechts vom Buch mit Nummer 1 an richtiger Position steht
  7. fahre mit Buch Nummer 3 fort
  8. usw.
- Offenbar kommen dadurch alle Bücher einmal an ihren richtigen Platz



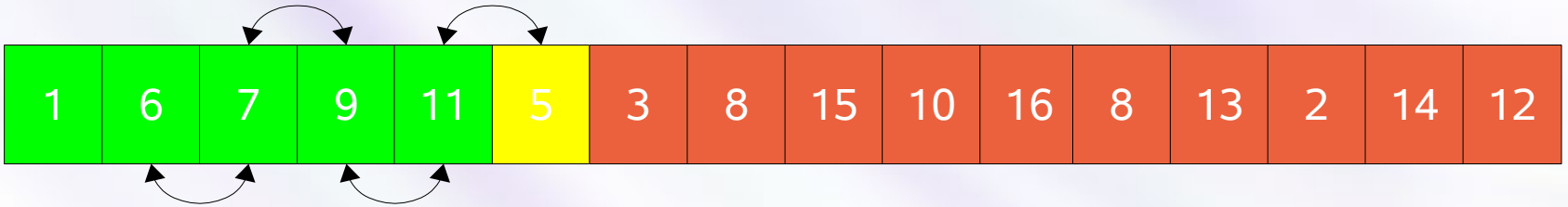
Programmieren

- Wie kann man das programmieren?
  - Feld von Zahlen
    - in der Informatik sagt man dazu Array
  - Werte durch einen Zahlenwert ausgewählt
    - Index
  - `A[i]` ist Wert an *i*-ter Stelle
    - auch: Wert unter Index *i*
  - In Mathematik schreibt man A_i
  - Feld der Länge *n*
  - Werte sind `A[1]`, `A[2]`, ..., `A[n-1]`, `A[n]`



Ein Programm

```
for i:=2 to n do                    /* Bücher an Stellen i=2 bis n */
   for j:=i downto 2 do             /* vergleich A[j] j=i, i-1, ... 2 mit A[j-1]*/
     if A[j-1] > A[j] then          /* A[j] steht falsch*/
     begin                          /* vertausche A[j-1] mit A[j] */
         h:=A[j];                   /* h speichert A[j] zwischen */
         A[j]:=A[j-1];
         A[j-1]:=h;
     end
   else break for j;                /* Ende der inneren j-Schleife */
```



Wie lange dauert es?

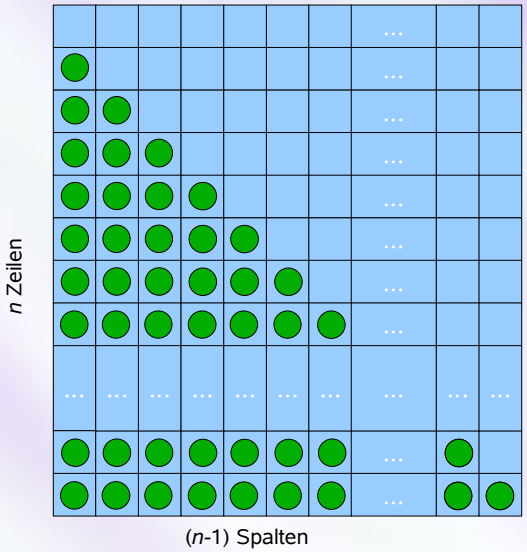
- Wie lange dauert das Aufräumen?
  - schlechtester Fall:
    alle Bücher genau verkehrt herum sortiert
  - unser Algorithmus
    - fängt von links an
    - vertauscht zweites Buch mit erstem
    - Vertauscht drittes Buch mit ersten beiden
    - Vertauscht viertes Buch mit ersten dreien
    - usw. Die Arbeit ist also $1+2+3+\ldots+n-1=\frac{n\cdot(n-1)}{2}$.

Ein bisschen Mathematik

- Die Formel

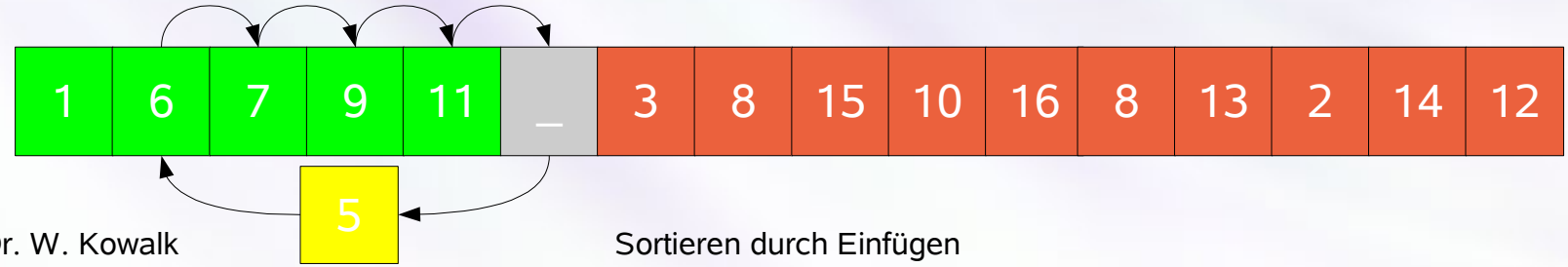
$$1+2+3+\ldots+n-1=\frac{n\cdot(n-1)}{2}.$$

lässt sich durch das folgende Bild einsehen



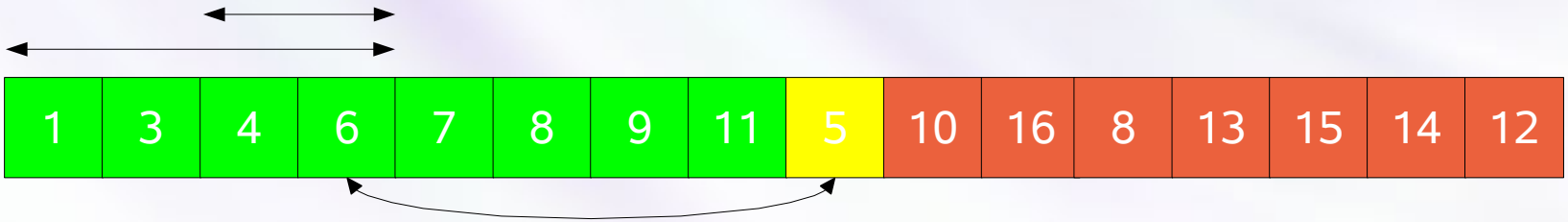
Verbesserungen

- Verbesserungen
  - Sortierverfahren zu umständlich
    - benachbarte Bücher werden immer vertauscht
  - Besser
    1. Nehme ein Buch heraus
    2. Schiebe alle Bücher nach rechts, bis zur Einfügestelle
    3. stellen das neue Buch an richtige Stelle
    - Statt zwei Bücher k-mal zu vertauschen
      (= 3·k-mal verschieben),
      verschieben wir (k+2)-mal ein Buch



Verbesserung: Binäres Suchen

- Weitere Verbesserungen
  - Finde Einfügestelle durch Binäres Suchen (siehe Algorithmus der letzten Woche)
    1. Teile sortierte Bücher in zwei Hälften (1-6, 7-11)
    2. suche links, wenn Einfügestelle kleiner als mittleres Buch ($1 < 5 \leq 6$)
       - sonst suche rechts weiter
    3. halbiere das kleinere Intervall (1-3, 4-6)
    4. usw.
  - Anzahl der Vergleiche 'logarithmisch'



Zusammenfassung

- Sortieren kann sehr schnell erledigt werden
  - wenn man nur den richtigen Algorithmus kennt
- in der Informatik wichtig, richtigen Algorithmen zu kennen und einzusetzen
  - richtiger Algorithmus macht Problem erst lösbar
  - weitere Verbesserungen dieses Algorithmus möglich
  - 'Höhere' Sortierverfahren noch schneller