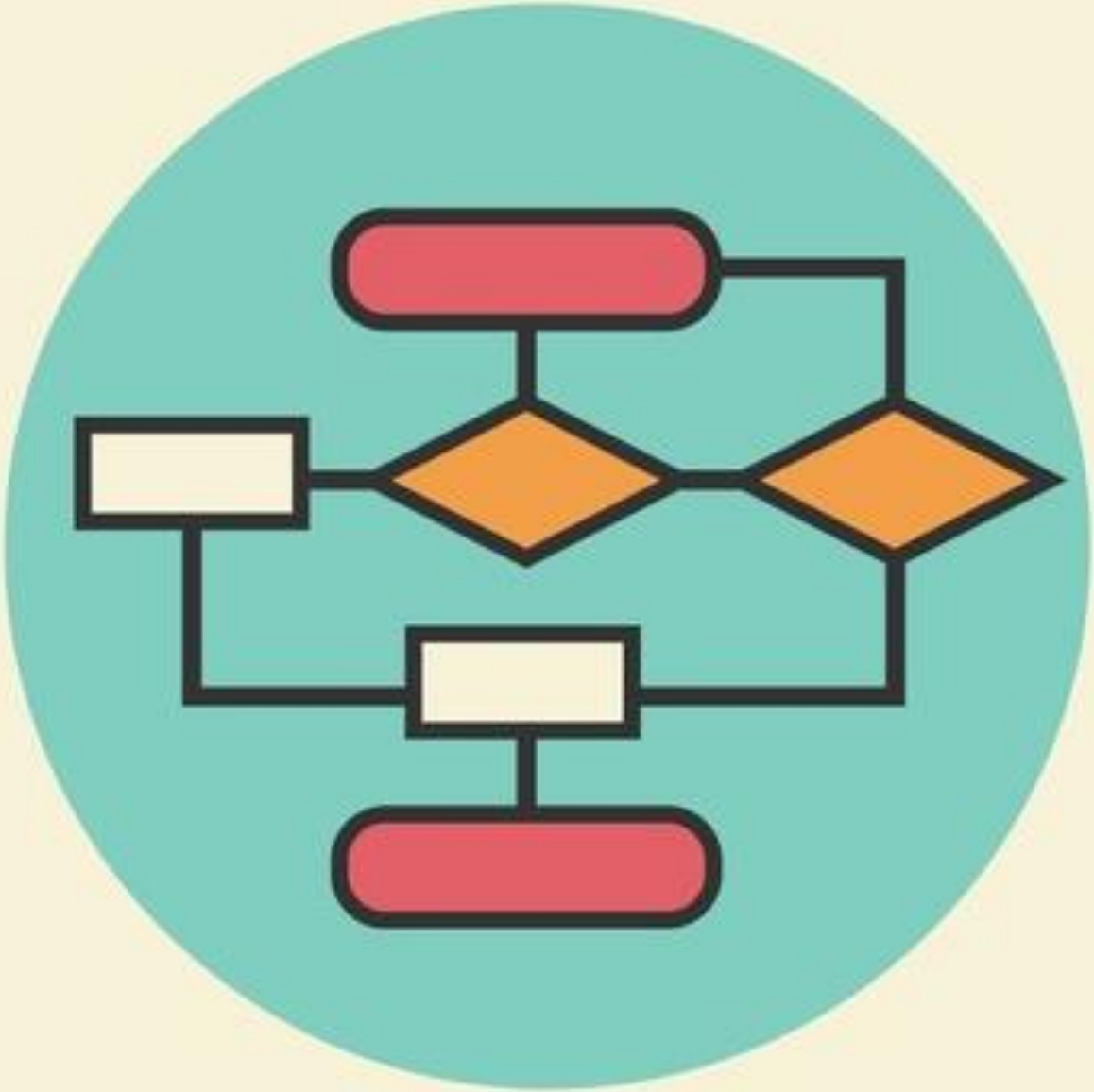
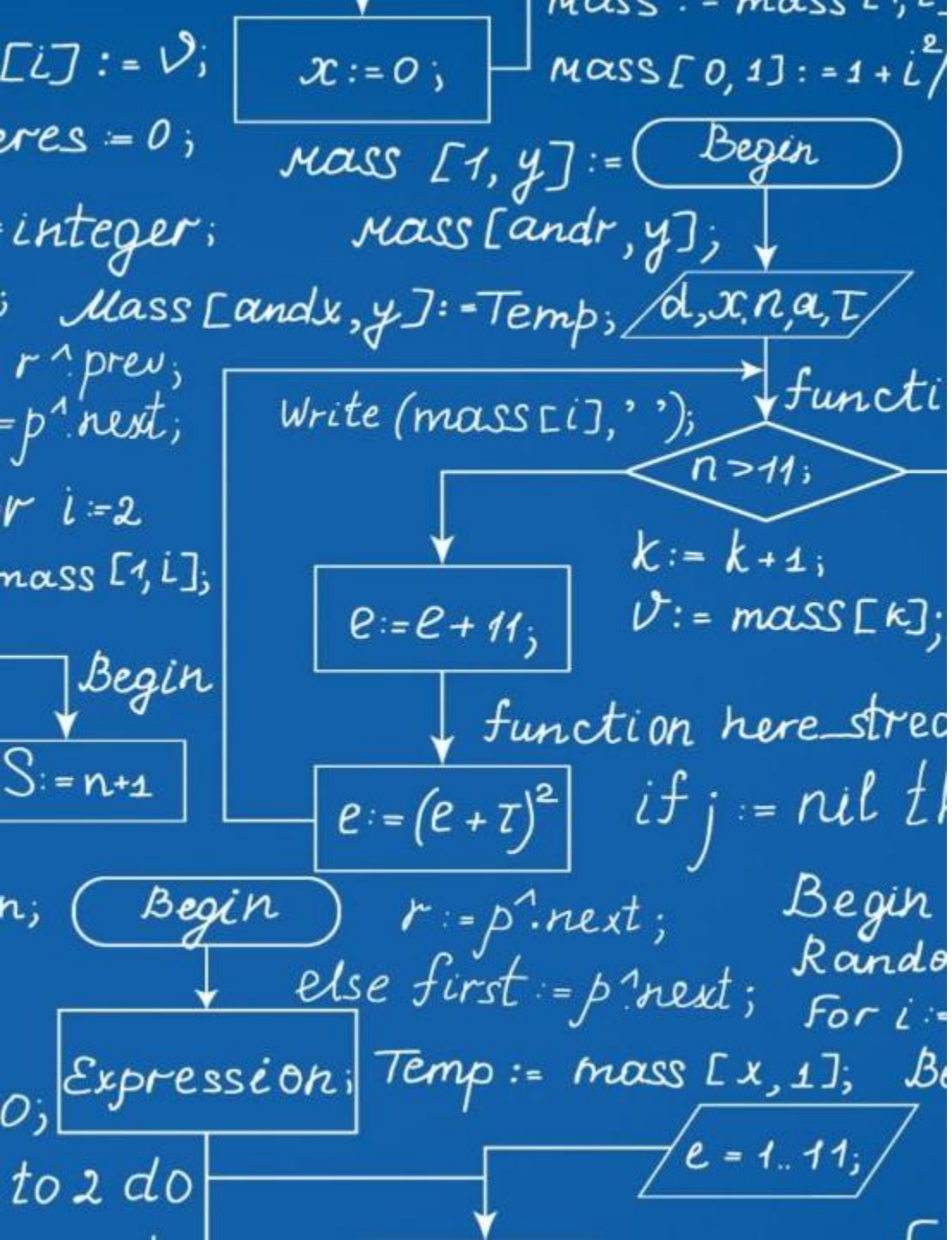
ALGORYTM SZYBKIEGO SORTOWANIA

Kacper Stanowicki





PLAN PREZENTACJI

- Gdzie jest klucz?
- Jak podzielić pole?
- Algorytm szybkiego sortowania

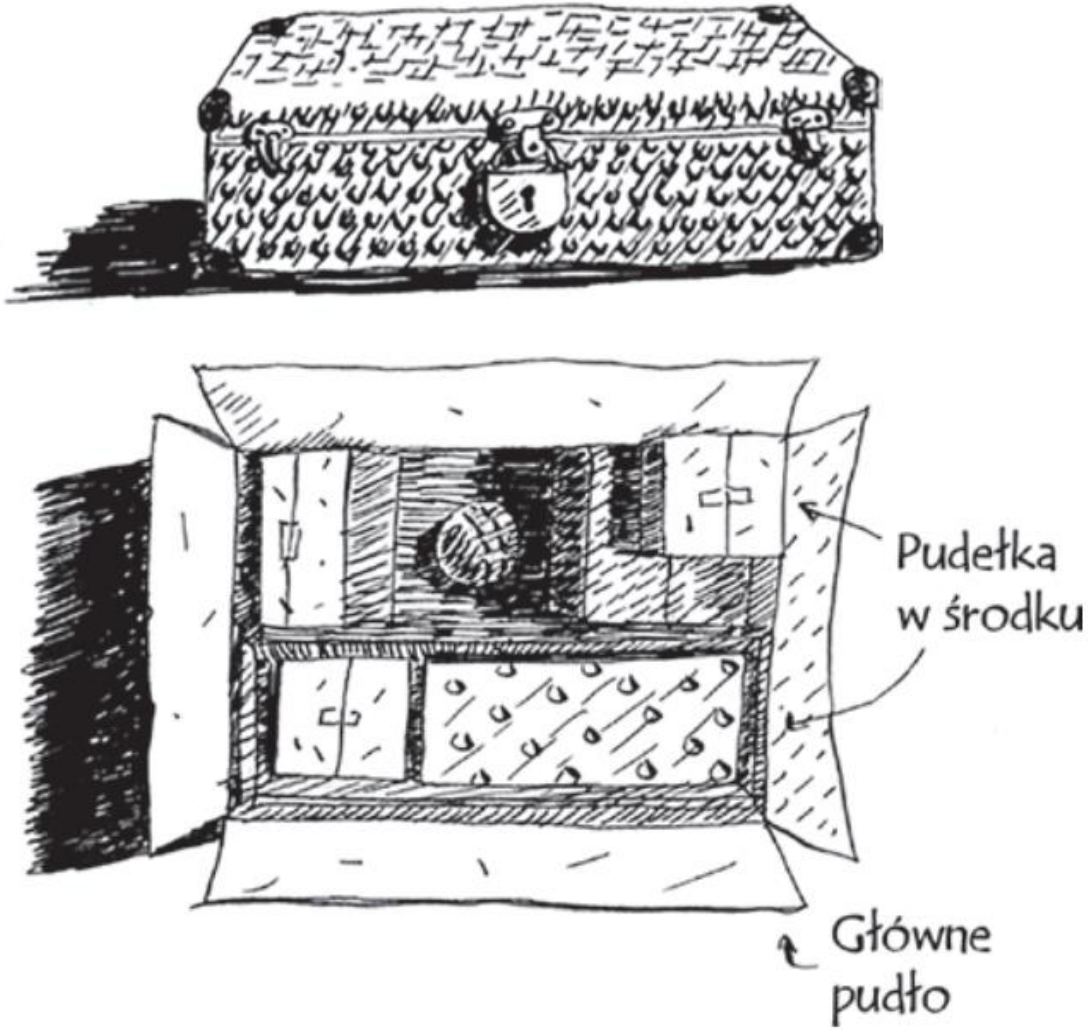
GDZIE JEST KLUCZ?

Wchodzisz do babci na strych i znajdujesz tajemniczą zamkniętą walizkę. Rozmyślając o tym co może się znajdować w tej walizce, biegniesz do babci zapytać się czy wie gdzie jest do niej klucz.

Babcia odpowiada: „znajduje się on w tym oto pudełku.”

A w nim znajdują się jeszcze inne pudełka, a możliwe że w nich kolejne itd.

Jak zorganizować prace poszukiwawcze?



GDZIE JEST KLUCZ? – PĘTLA WHILE


```
def look_for_key(main_box):
  pile = main_box.make_a_pile_to_look_through()
  while pile is not empty:
    box = pile.grab_a_box()
    for item in box:
      if item.is_a_box():
        pile.append(item)
      elif item.is_a_key():
        print "Znaleziono klucz!"
```

GDZIE JEST KLUCZ? - REKURENCJA


```
def look_for_key(box):
  for item in box:
    if item.is_a_box():
      look_for_key(item)
    elif item.is_a_key():
      print "Znaleziono klucz!"
```

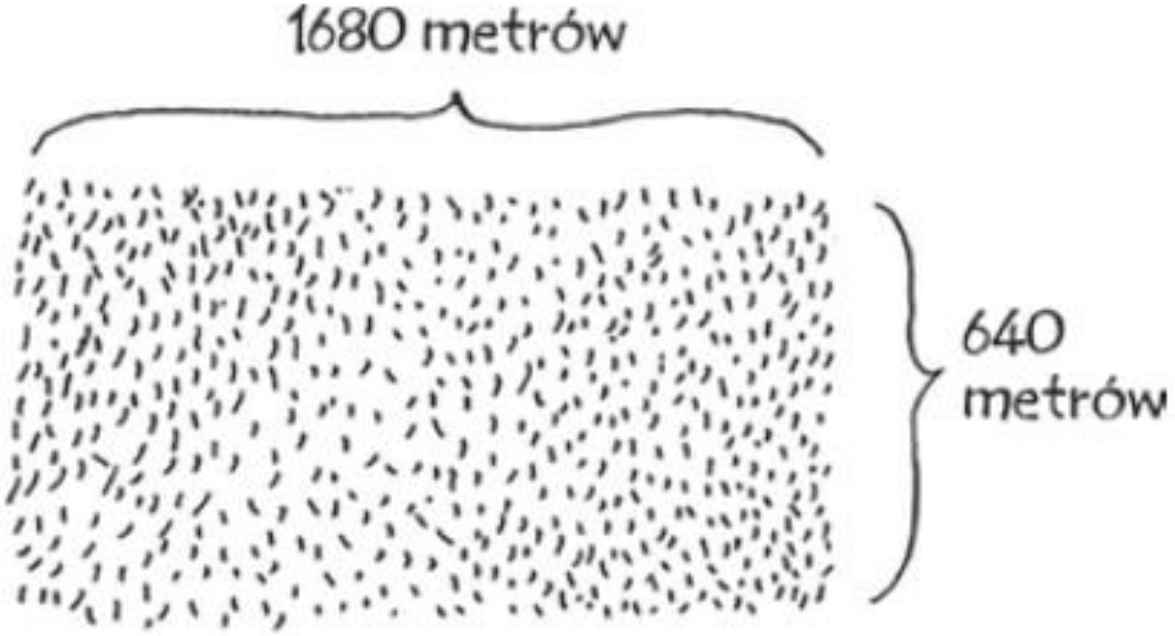
Przypadek rekurencyjny

Przypadek podstawowy

JAK PODZIELIĆ POLE?

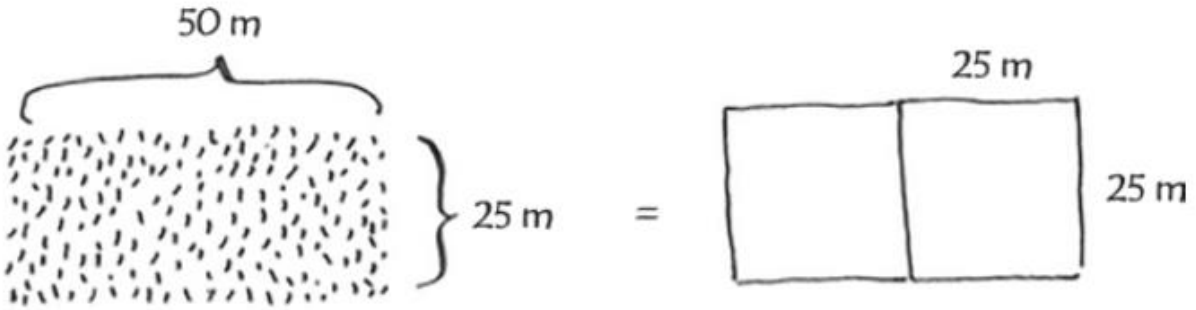
Pewien rolnik prosi cię o pomoc w podzieleniu jego pola na jak największe równe kwadraty.

Jak możemy znaleźć największy możliwy rozmiar kwadratu?

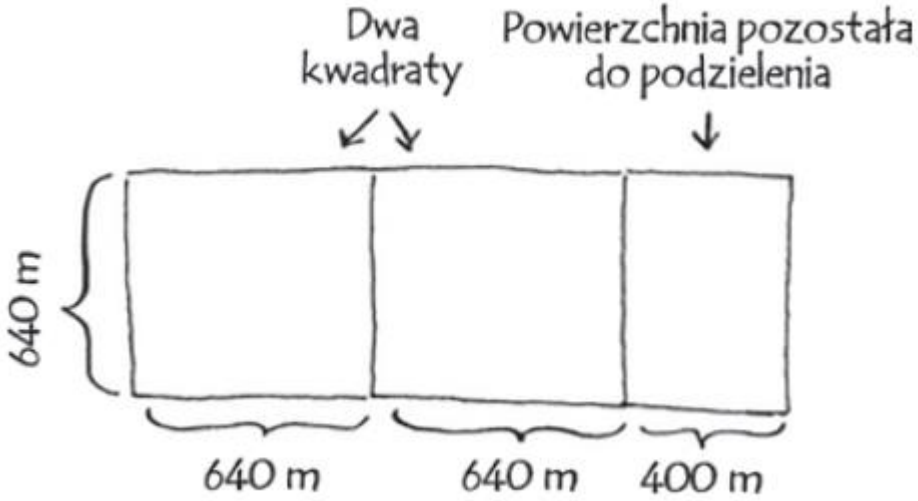


JAK PODZIELIĆ POLE?– DZIEL I RZĄDŹ

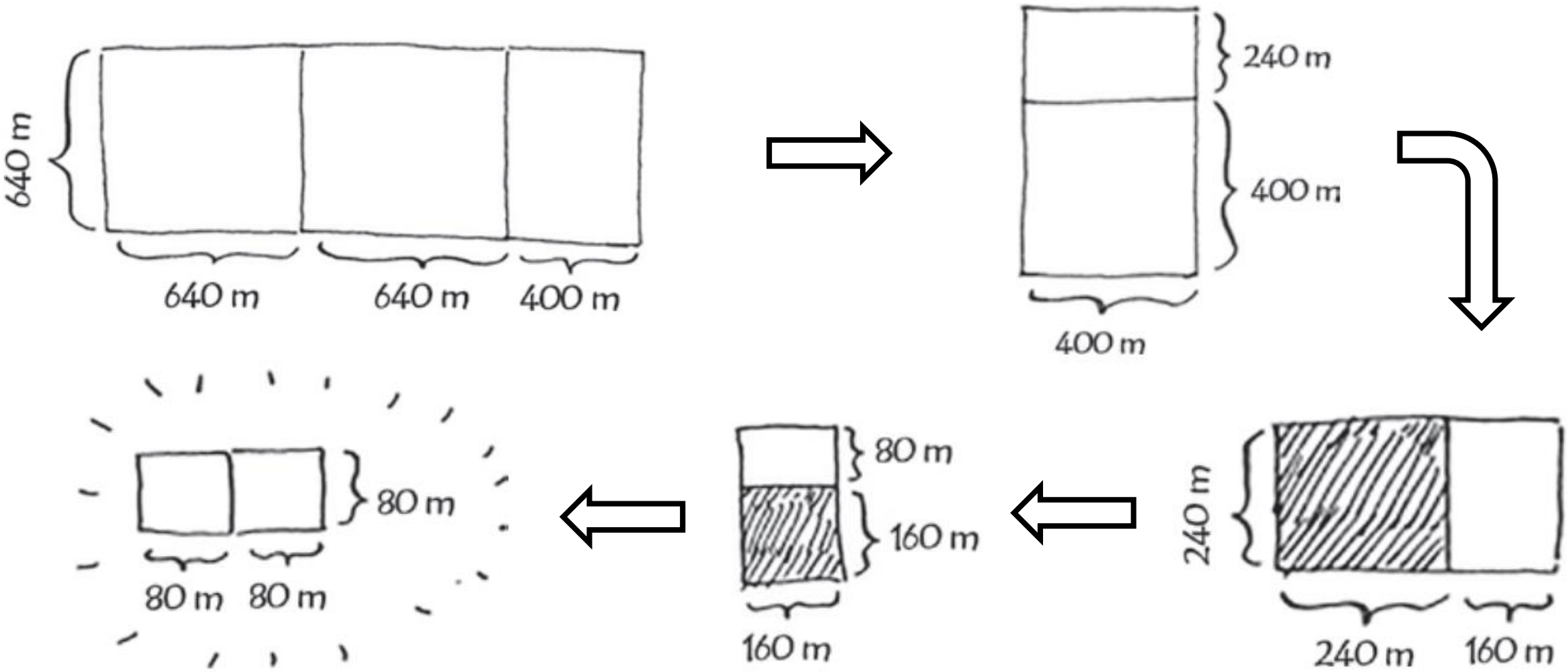
Przypadek podstawowy



Przypadek rekurencyjny

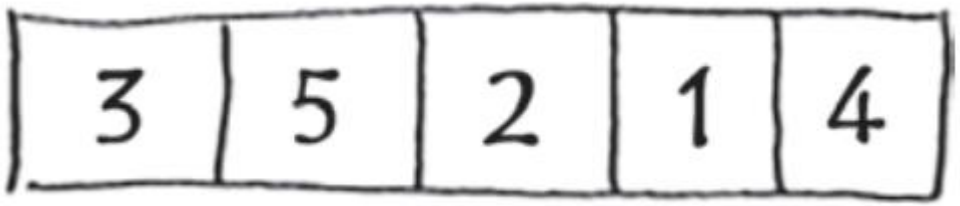


JAK PODZIELIĆ POLE?– DZIEL I RZĄDŹ



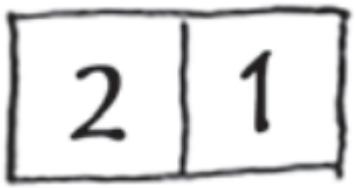
ALGORYTM SZYBKIEGO SORTOWANIA

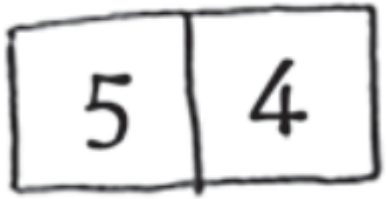
Tablica do posortowania:



Element osiowy

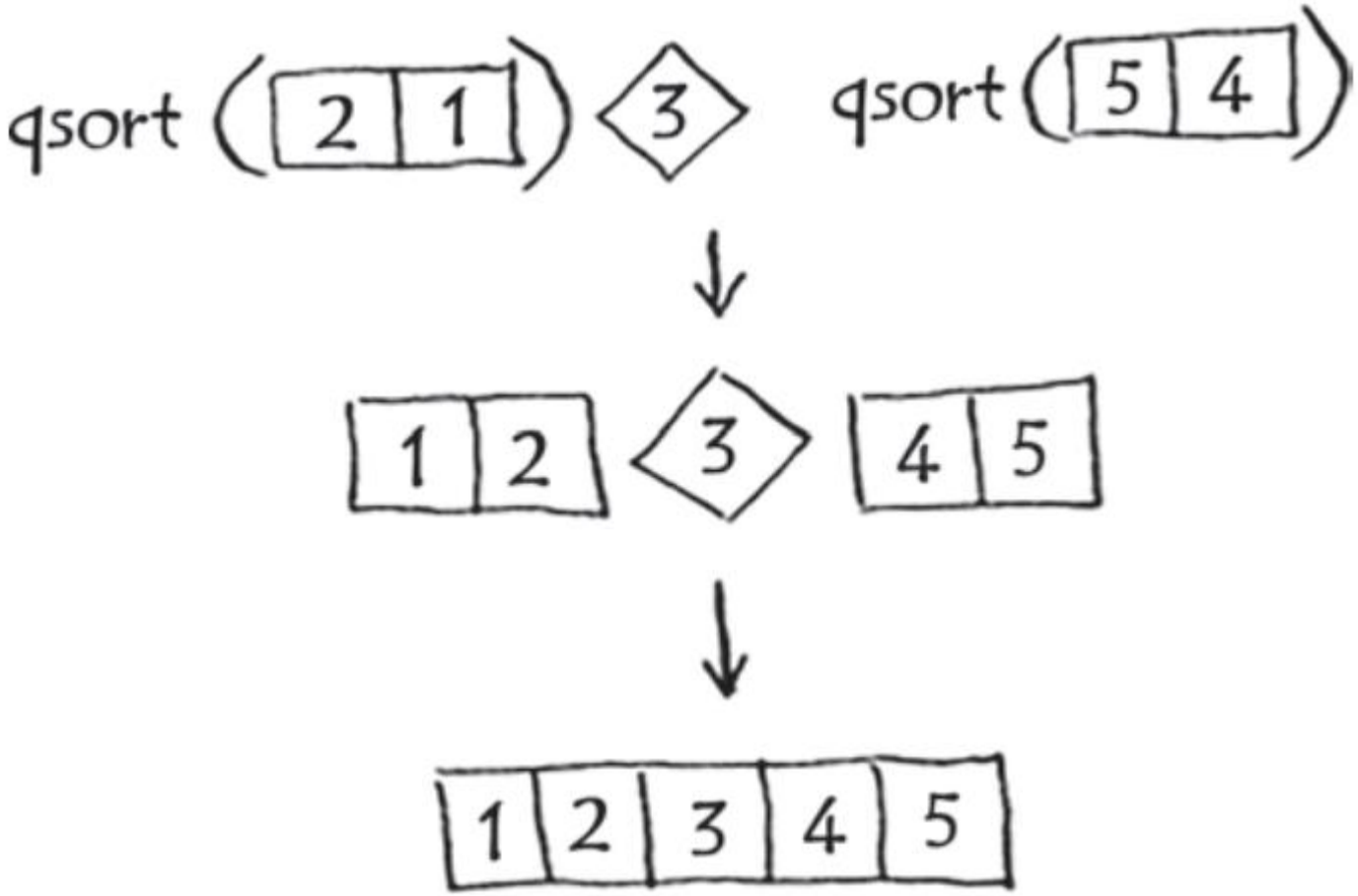
Podtablica z elementami <= 3





Podtablica z elementami > 3

ALGORYTM SZYBKIEGO SORTOWANIA



ALGORYTM SZYBKIEGO SORTOWANIA

```
def quicksort(array):
  if len(array) < 2:
    return array
  else:
    pivot = array[0]
    less = [i for i in array[1:] if i <= pivot]

    greater = [i for i in array[1:] if i > pivot]

    return quicksort(less) + [pivot] + quicksort(greater)
```

Przypadek podstawowy: tablice puste i jednoelementowe są z góry „posortowane”.

Przypadek rekurencyjny.

Podtablica zawierająca wszystkie elementy mniejsze od elementu osiowego.

Podtablica zawierająca wszystkie elementy większe od elementu osiowego.

LITERATURA

- Algorytmy. Ilustrowany przewodnik - Aditya Bhargava
- https://www.techidence.com/whats-an-algorithm-understand-how-it-works-in-apps-and-websites/
- http://codeprogramming.org/2020/06/05/why-are-algorithms-so-important-in-programming/