

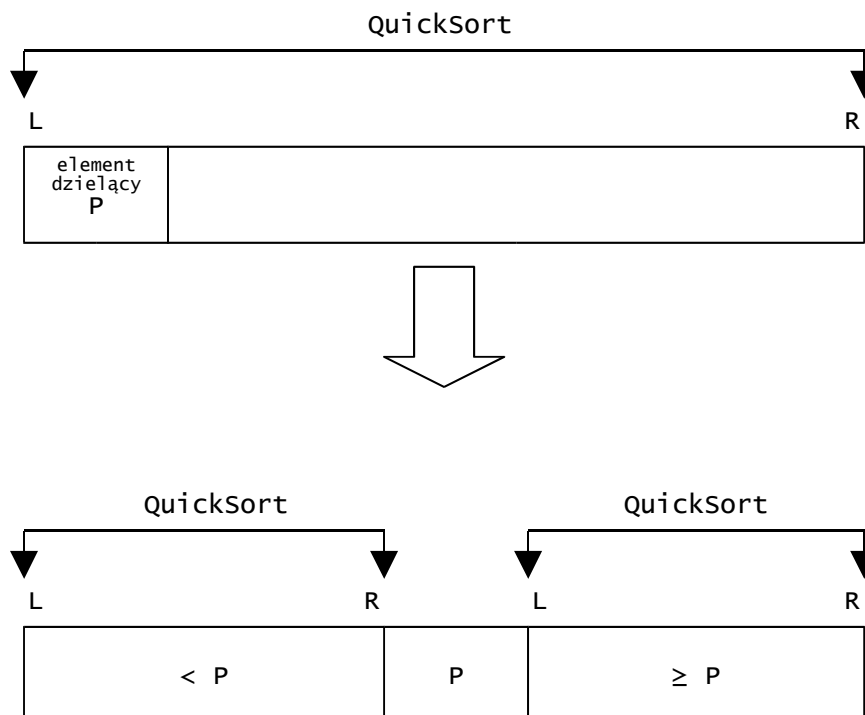
## Sortowanie szybkie (QuickSort)

Algorytm *QuickSort*, znany od roku 1960 (C.A.R. Hoare), jest jednym z najczęściej używanych algorytmów sortowania. Algorytm jest uważany za najszybszy dla losowych danych wejściowych.

Algorytm sortowania szybkiego działa na zasadzie podziału tablicy na dwie części: jedna zawiera elementy o mniejszych wartościach, a druga elementy o większych wartościach. *Elementem dzielącym* (oś, trzpień, ang. *pivot*) obie części tablicy jest jeden z jej elementów. Byłoby najlepiej, gdyby w jego charakterze wybierać tzw. *medianę*. Jest nią element mniejszy od połowy (lub równy połowie) i większy od drugiej połowy (lub równy drugiej połowie) wszystkich elementów tablicy. Idea algorytmu jest następująca:

- 1) wybieramy losowo jeden element tablicy (np. pierwszy element),
- 2) dzielimy tablicę na dwie grupy: elementy mniejsze i elementy większe lub równe względem wybranego elementu,
- 3) sortujemy obie grupy osobno.

Rys. 1. Zasada działania procedury QuickSort



Procedura *QuickSort* będzie zatem miała następującą konstrukcję:

1. Jako element osiowy przyjmij element lewy ( $tab[L]$ ).
2. Początkowa pozycja elementu osiowego:  $m = L$ .
3. Przeszukaj pozostałą część tablicy (od  $i = L + 1$  do  $R$ ); jeśli znajdziesz element o wartości mniejszej niż element osiowy ( $tab[i] < tab[L]$ ):
  - a) zwiększ indeks  $m$  (poszukiwany indeks komórki, w której umieścimy element osiowy);
  - b) zamień miejscami komórki  $tab[i]$  oraz  $tab[m]$ .
4. Wstaw element osiowy pod indeks  $m$  (zamiana miejscami komórek  $tab[L]$  i  $tab[m]$ ).
5. Jeśli jest jeszcze fragment tablicy nieposortowany po lewej lub po prawej stronie elementu osiowego (znajduje się on teraz w komórce o indeksie  $m$ ), wywołaj procedurę *QuickSort* dla tego fragmentu.

Aby program był w pełni funkcjonalny, dodajmy do niego procedurę generowania elementów losowych w tablicy (*GenerujTablice*), procedurę zamiany miejscami dwóch elementów (*Zamiana*) oraz procedurę wypisywania wszystkich elementów tablicy (*WypiszTablice*).

Oto pełny kod programu:

```
program Sort;

const
  MAX = 10-1;
  ZAKRES = 1000;
var
  tab: array[0..MAX] of Integer;
  i: integer;

{ Uzupełnianie tablicy losowymi wartościami }
procedure GenerujTablice;
var
  i: integer;
begin
  for i:=0 to MAX do tab[i]:= Random(ZAKRES+1);
end;

{ wypisywanie wszystkich elementów tablicy }
procedure wypiszTablice;
var
  i: integer;
begin
  for i:=0 to MAX do write(tab[i]:8);
end;

{ Procedura zamiany dwóch elementów miejscami }
procedure Zamiana(var a, b: integer);
var
  z: integer;
begin
  z:=a; a:=b; b:=z;
end;
```

```

{ Procedura sortowania - w nawiasach klamrowych podano główne punkty
  algorytmu}
procedure QuickSort(var tab: array of integer; left, right:integer);
var
  i, m, z: integer;
begin
  m:=left;                               {2}
  for i:=left+1 to right do              {3}
    if tab[i]<tab[left] then
      begin
        Inc(m);                           {3a}
        Zamiana(tab[m], tab[i]);          {3b}
      end;
    Zamiana(tab[left], tab[m]);           {4}
    if left < m-1 then quicksort(tab, left, m-1); {5}
    if m+1 < right then quicksort(tab, m+1, right);
  end;

{ Główna część programu }
begin
  writeln;
  Randomize;
  GenerujTablice;
  { wywołanie procedury QuickSort dla całej tablicy }
  QuickSort(tab,0,MAX);
  wypiszTablice;
  writeln('Press Enter..');
  ReadLn;
end.

```