

Algorytmy sortowania

1. Problem sortowania:

Dane wejściowe: ciąg n liczb $\langle a_0, a_1, \dots, a_n \rangle$

Wynik: permutacja $\langle a'_0, a'_1, \dots, a'_n \rangle$ ciągu wejściowego taka że $a'_0 \leq a'_1 \leq a'_n$

2. Sortowanie bąbelkowe ($O(n^2)$):

przebieg po całym ciągu i porównanie dwóch kolejnych elementów i zamiana ich kolejności w przypadku nieporządku

```
procedure bubbleSort( A : lista elementów do posortowania )
  n = liczba_elementów(A)
  do
    for (i = 0; i < n-1; i++) do:
      if A[i] > A[i+1] then
        swap(A[i], A[i+1])
      end if
    end for
    n = n-1
  while n > 1
end procedure
```

3. Sortowanie szybkie $\Theta(n \log n)$ / $O(n^2)$:

wybór elementu podziału i podzielenie reszty elementów względem mniejsze i większe od niego i rekursyjne wywołanie całości na powstałych podciągach

QUICKSORT(A, p, r)

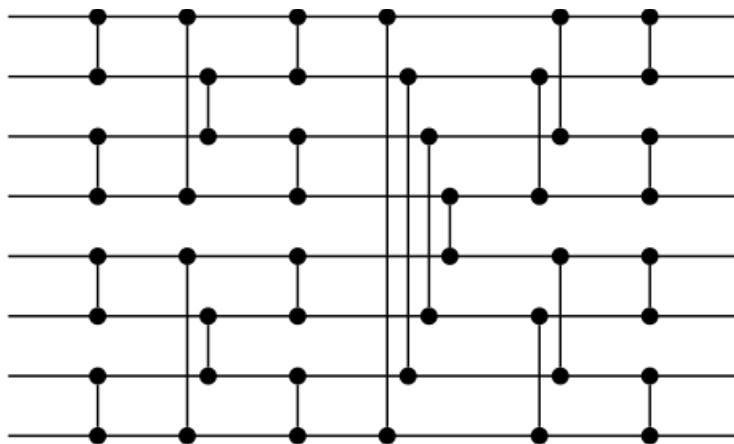
```
1  if  $p < r$ 
2     $q = \text{PARTITION}(A, p, r)$ 
3    QUICKSORT( $A, p, q - 1$ )
4    QUICKSORT( $A, q + 1, r$ )
```

PARTITION(A, p, r)

```
1   $x = A[r]$ 
2   $i = p - 1$ 
3  for  $j = p$  to  $r - 1$ 
4    if  $A[j] \leq x$ 
5       $i = i + 1$ 
6      exchange  $A[i]$  with  $A[j]$ 
7  exchange  $A[i + 1]$  with  $A[r]$ 
8  return  $i + 1$ 
```

4. Bitonic sort:

równoległy algorytm sortowania oparty o sieć sortującą



5. Map/Reduce Sort:

Równoległy, przydatny przy przetwarzaniu olbrzymich ilości danych, pracuje na wartościach typu klucz-wartość.

Etapy działania algorytmu:

- mapowanie – rozdzielenie wejścia na grupy, według konkretnej cechy obiektów wejściowych (np. kolor przy sortowaniu kart do gry)
- shuffle/sort – etap pośredni wymieniający dane z grup między jednostkami i sortowanie w nich
- redukcja – przetworzenie i połączenie grup
-

Działanie: podzielenie dużej ilości danych na segmenty i przekazanie każdego z nich do mapperów, każdy mapper przetwarza swoje dane (sortuje je) a potem przekazuje do reduktora. On mając posortowane fragmenty danych łączy je w jedną posortowaną całość.

6. Inne:

- sortowanie przez zliczanie: zliczenie ilości elementów mniejszych od aktualnego
- sortowanie przez wstawianie: budowanie ciągu posortowanego dokładając po jednym elemencie z ciągu wejściowego i wstawienie go w odpowiednie miejsce w ciągu wyjściowym
- sortowanie przez scalanie: rekurencyjne sortowanie metodą dziel i zwyciężaj
- sortowanie koktajlowe: idąc z jednego końca ciągu na drugi bierzemy największy element a wracając najmniejszy
- sortowanie przez wybór: wybranie z pozostałych elementów najmniejszego i wstawienie go w aktualny indeks
- bogo-sort: losowanie permutacji ciągu aż trafimy na posortowany