

Celem laboratorium nr 01 jest zapoznanie się z algorytmami dotyczącymi znajdowania wartości minimalnej/maksymalnej w zbiorze, sortowaniem i wyszukiwaniem.

Zapoznajcie się z poniższymi algorytmami i przygotujcie ich implementację dla zbioru danych (na początek może być tablica liczb całkowitych).

Pamiętajcie, że macie to zrealizować algorytmicznie (korzystając z pętli, instrukcji sterujących, etc), a nie „językowo” (użycie `sorted()` czy `find()`).

1. Znajdywanie wartości minimalnej/maksymalnej.

W przypadku danych liczbowych, znalezienie minimum lub maksimum nie stanowi problemu. Jednakże, gdyby rozważyć zbiór np. ciągów znaków lub jakichś struktur zawierających więcej niż jedną zmienną, pojawia się pewien problem. Należy wpierw ustalić co oznacza $A > B$ w przypadku np. ciągów znaków A i B. Czy chodzi o ich długości? Może o sumę kodów ASCII każdego ze znaków? Czy może o liczbę słów (rozdzielonych białym znakiem) w tekście?

W ramach zajęć rozważcie prostą tablicę liczb całkowitych (najlepiej wylosować je raz i zapisać gdzieś w pliku) i przygotujcie algorytm, który znajdzie wartość minimalną/maksymalną.

2. Algorytmy sortowania.

Podobnie jak w przypadku podpunktu 1., należy wpierw ustalić co oznacza że A jest większe od B? Na potrzeby zajęć przyjmijmy uproszczoną wersję, w której wciąż bazujemy na liczbach pseudolosowych (dlaczego nie losowych? O tym kiedy indziej 😊) z podpunktu 1.

Naszym celem będzie implementacja podstawowych algorytmów sortowania oraz analiza ich czasu działania.

Skorzystamy do tego celu z poniższego kodu.

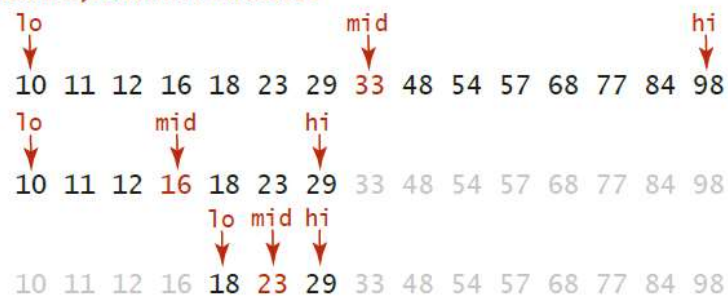
```
long mStart = System.currentTimeMillis();  
// wykonanie programu  
long executionTime = System.currentTimeMillis() - mStart;
```

3. Wyszukiwanie binarne (na posortowanym zbiorze).

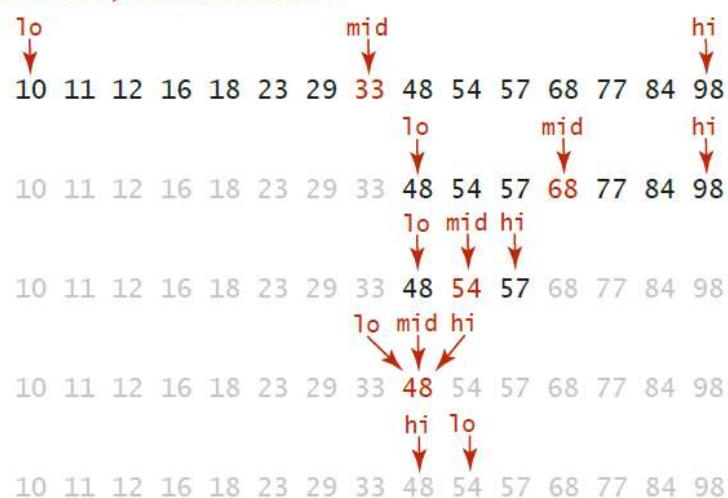
Wyszukiwanie binarnemu wymaga, aby przeszukiwana tablica była już posortowana. Zaletą wyszukiwania binarnego jest to, że przeszukiwanie n-elementowej tablicy zajmuje w nim tylko $O(\log n)$ czasu.

Algorytm jest realizowany metodą "dziel i zwyciężaj". Dzieli on tablicę na mniejsze podtablice do momentu wyszukania pozycji (lub nie w przypadku gdy taki element nie istnieje) elementu szukanego.

Udane wyszukiwanie wartości 23



Nieudane wyszukiwanie wartości 50



Wyszukiwanie binarne w posortowanej tablicy