

## Algorytmy i struktury danych w bibliotekach C++

- Należy zapoznać się z algorytmami i strukturami danych z biblioteki STL w C++ i używać ich przy rozwiązywaniu poniższych zadań.
- Informacje na ten temat można znaleźć pod adresami <http://www.cplusplus.com/reference/algorithm/> i <http://www.cplusplus.com/reference/stl/>
- Powyższe zadania należy wykonać z użyciem klas i funkcji bibliotecznych

1) Mamy zadeklarowaną tablicę liczb:

```
int liczby[] = {32, 71, 12, 45, 26, 80, 53, 32};
```

- a) posortuj liczby najpierw malejąco, a potem rosnąco
- b) sprawdź czy element  $x$  występuje w tablicy
- c) oblicz liczbę elementów w tablicy należących do przedziału  $\langle a, b \rangle$

2) Mamy dane dwie posortowane tablice liczb oraz puste wektory liczb:

```
int liczby1[] = {12, 26, 32, 32, 45, 53, 71, 80};  
int liczby2[] = {13, 18, 44, 53, 55, 75, 88, 99};  
vector<int> wspolne(8);  
vector<int> wynik(16);
```

- a) scal tablice w jeden ciąg posortowany i umieść go w kontenerze *wynik*.
- b) wyznacz część wspólną z obu tablic liczb i umieść ją w kontenerze *wspolne*.

3) W zadaniu nr 2 z listy zadań nr 3 (programowanie dynamiczne i algorytmy zachłanne) Jaś lepił kule śnieżne tak, by zmęczyć się jak najmniej. Wiadomo, że należało tam zlepić ze sobą zawsze dwie kulki o najmniejszej wadze. Oto tablica z wagami kul:

```
int kulki[] = {5, 4, 1, 8, 3};
```

Napisz kod, który rozwiąże nam to zadanie korzystając z odpowiedniej struktury danych (wskazówka – kolejka priorytetowa).

4) Na przykładach z biblioteki c++ pokaż użycie następujących struktur danych:

- a) zbiór uporządkowany (zrównoważone drzewo binarne):
  - dodaj/usuń wyszukaj kilka elementów
  - wypisz wszystkie elementy w porządku rosnącym
  - wypisz elementy z określonego przedziału (np. od 100 do 200)
- b) zbiór nieuporządkowany (tablica haszująca):
  - dodaj/usuń wyszukaj kilka elementów
  - ustaw współczynnik zapelnienia tablicy haszującej