

Laboratorium 1

Tablice

1. Zaimplementuj algorytm wyszukiwania liniowego określonej wartości w tablicy jako metodę: `bool find (long key);`
2. Zaimplementuj algorytm wyszukiwania binarnego jako metodę:
`bool findbin (long key);` Uwaga! Pamiętaj, że w algorytmie zakłada się, że wejściowa tablica jest uporządkowana.
3. Zaprojektuj i zaimplementuj algorytm, który odwróci tablicę jako metodę:
`void reverse ();`
4. Zaimplementuj algorytm wyszukiwania elementu maksymalnego w tablicy jako metodę: `long max();`
5. Zaimplementuj algorytm wyszukiwania elementu minimalnego w tablicy jako metodę: `long min();`
6. Zaprojektuj i zaimplementuj algorytm, który usuwa określoną wartość z tablicy jako metodę: `void remove(long value);`
7. Zaprojektuj i zaimplementuj algorytm, który liczy średnią arytmetyczną elementów nieujemnych w tablicy jako metodę: `long average();`
8. Zaprojektuj i zaimplementuj algorytm, który zlicza liczbę elementów parzystych w tablicy jako metodę: `int even();`
9. Zaprojektuj i zaimplementuj algorytm, który zlicza liczbę elementów nieparzystych w tablicy jako metodę: `int odd();`
10. Zaprojektuj i zaimplementuj algorytm, który zlicza liczbę wystąpień elementu `key` w tablicy jako metodę: `int numberInstances(long key);`
11. Stwórz kolejkę wykorzystując tablicę jako strukturę danych. Zrealizuj operację odczytu pierwszej i ostatniej wartości. Zrealizuj funkcję `pop` pobierającą wartość zgodnie z LIFO oraz zgodnie z FIFO.

Stos

1. Zaprojektuj realizację stosu z operacjami: odkładania na stos `push (int value);` zdejmowania ze stosu `int pop();` sprawdzania zajętości stosu: `int isEmpty(); isEmpty(); int getSize()` zwracającą rozmiar stosu, `int Index();` zwracającą numer aktualnej pozycji w stosie

Listy

1. Zaprojektuj realizację listy jedno i jako uogólnienie dwukierunkowej. Zdefiniuj operacje przeglądania listy, dodawania i usuwania elementów w liście