

WYMAGANIA EDUKACYJNE – INFORMATYKA

zakres rozszerzony • klasa 3

Ocenę celującą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- pisze programy o wysokim stopniu trudności lub oznaczone trzema gwiazdkami w podręczniku,
- implementuje w języku C++ algorytm Euklidesa, stosując iterację i rekurencję,
- pisze programy szyfrujące i deszyfrujące z wykorzystaniem zaawansowanych szyfrów (np. permutacyjny lub Vigenere'a) i różnych kluczy,
- implementuje w języku C++ algorytm wyszukiwania binarnego w wersji rekurencyjnej,
- pisze programy sortujące dane różnego typu w plikach tekstowych (liczby, napisy, pary),
- stosuje zaawansowane algorytmy i struktury danych do wyszukiwania spójnych podciągów,
- pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję

Ocenę bardzo dobrą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania,
- pisze programy o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku,
- optymalizuje rozwiązania,
- stosuje zaawansowane funkcje środowiska i języka programowania (np. z biblioteki STL),
- dobiera struktury danych i metody do rodzaju problemu,
- w programach wykonujących działania na liczbach w różnych systemach pozycyjnych wykorzystuje bibliotekę string i strukturalne typy danych,
- wykonuje operacje arytmetyczne na liczbach w różnych systemach, implementuje je w języku C++,
- wykorzystuje algorytm Euklidesa do działań na ułamkach, stosując struktury lub pary (typ pair),
- szyfruje dane wczytane z pliku z uwzględnieniem polskich znaków diakrytycznych,
- pisze program odczytujący informację ukrytą za pomocą szyfru Cezara z wykorzystaniem analizy częstości znaków w tekście,
- szacuje złożoność obliczeniową programów sortujących, modyfikuje funkcje sortujące, zmieniając porządek sortowania,
- wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach,
- optymalizuje program realizujący algorytm sita Eratostenesa i szacuje jego złożoność czasową,
- pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję,
- do implementacji rozszerzonego algorytmu Euklidesa stosuje zarówno iterację, jak i rekurencję,

Ocenę dobrą otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- określa specyfikację algorytmu (dane, wynik),
- pisze programy o różnym stopniu trudności, szacuje ich efektywność,
- przedstawia omawiane algorytmy w postaci opisu słownego, listy kroków, pseudokodu,
- dobiera typy danych do realizacji problemu,
- implementuje w języku C++ algorytmy wykonujące działania arytmetyczne na liczbach w różnych systemach,
- stosuje w programach algorytm Euklidesa do obliczenia NWD i NWW,
- wykorzystuje algorytm Euklidesa do działań na ułamkach,
- szyfruje dane wczytane z pliku tekstowego,
- implementuje w języku C++ algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic,
- stosuje algorytm wyszukiwania binarnego i oszacowuje jego złożoność czasową,
- pisze programy sortujące metodami prostymi z zastosowaniem funkcji typu void,
- pisze program realizujący algorytm sita Eratostenesa,
- implementuje w języku C++ algorytmy wyszukujące spójne podciągi o różnych cechach,
- porównuje algorytmy iteracyjne i rekurencyjne (liczbę wykonywanych operacji), szacuje ich złożoność czasową,
- zapisuje w postaci programu rozszerzony algorytm Euklidesa, wyjaśnia jego działanie i zastosowanie,

Ocenę dostateczną otrzymuje uczeń, który spełnia kryteria oceny dopuszczającej, a ponadto:

- przedstawia krótkie algorytmy w postaci listy kroków, opisu słownego, pseudokodu,
- dodaje liczby binarne,
- wykonuje działania arytmetyczne na liczbach w systemach liczbowych o różnych podstawach,
- implementuje w języku C++ algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem,
- pisze program szyfrujący napis szyfrem Cezara,
- omawia algorytm zliczania znaków w tekście oraz wyszukujący maksimum z wykorzystaniem tablic,
- implementuje w języku C++ algorytmy wyszukiwania liniowego i liniowego z wartownikiem, porównuje ich efektywność,
- omawia algorytm sita Eratostenesa,

- przedstawia algorytmy znajdowania spójnych podciągów, wyznaczania najdłuższego z nich oraz podciągu o największej sumie elementów,
- implementuje w języku C++ algorytmy rekurencyjne: obliczanie elementów ciągu Fibonacciego, wartości silni i potęgi,
- omawia rozszerzony algorytm Euklidesa,

Ocenę dopuszczającą otrzymuje uczeń, który:

- definiuje podstawowe pojęcia z algorytmiki i programowania: algorytm, program, warunek, iteracja, rekurencja,
- korzysta ze środowiska programistycznego: pisze w nim kod, kompiluje i uruchamia program, odczytuje i zapisuje pliki,
- pisze programy o niewielkim stopniu trudności,
- omawia pojęcia: złożoność obliczeniowa algorytmu, algorytm naiwny, algorytm optymalny, złożoność pesymistyczna, złożoność oczekiwana (średnia),
- korzysta z podstawowych funkcji języka: operacji wejścia i wyjścia, instrukcji warunkowych i iteracyjnych, gotowych funkcji bibliotecznych,
- wymienia podstawowe typy danych, operacje arytmetyczne i logiczne,
- w pisanych programach korzysta ze strukturalnych typów danych: napisów, struktur, tablic,
- dodaje pisemnie liczby binarne,
- omawia geometryczną interpretację algorytmu Euklidesa,
- definiuje pojęcia: kryptologia, kryptografia, kryptoanaliza, tekst jawny, klucz, szyfrogram,
- rozróżnia szyfry podstawieniowe i przestawieniowe,
- omawia szyfr Cezara jako przykład szyfru podstawieniowego i szyfr kolumnowy jako przykład szyfru przestawieniowego,
- wyjaśnia, na czym polega łamanie szyfru,
- omawia algorytm zliczania znaków w tekście,
- wczytuje dane z pliku tekstowego, zapisuje wyniki w pliku,
- omawia algorytmy wyszukiwania liczby w zbiorach uporządkowanym i nieuporządkowanym,
- stosuje funkcję losującą w tworzonych programach,
- omawia metody sortowania prostego (bąbelkowe, przez wybieranie) na przykładowych danych,
- wypisuje liczby pierwsze z zadanego przedziału, stosując metodę sita Eratostenesa,
- wyszukuje w ciągu liczb spójne podciągi (nierosnący, niemalejący, stały), wskazuje najdłuższe, oblicza ich sumę,
- definiuje pojęcia iteracji i rekurencji,
- omawia zasadę złotego podziału,
- opisuje rozszerzony algorytm Euklidesa,