



Sortowanie bąbelkowe w języku C++

- [Wprowadzenie](#)
- [Animacja](#)
- [Przeczytaj](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Sortowanie bąbelkowe w języku C++

Źródło: Eliška Motisová, domena publiczna.

Znasz już różne algorytmy sortowania. Jednym z najprostszych algorytmów porządkujących dane jest algorytm sortowania bąbelkowego (*bubble sort*). Podstawowe informacje na temat tego zagadnienia zostały omówione w e-materiale [Sortowanie bąbelkowe](#). Tutaj natomiast zajmiemy się jego implementacją w języku C++.

Implementację algorytmu w innych językach programowania znajdziesz w e-materiałach:

- [Sortowanie bąbelkowe w języku Python](#)
- [Sortowanie bąbelkowe w języku Java](#).

Więcej zadań? Przejdź do: [Sortowanie bąbelkowe – zadania maturalne](#).

Twoje cele

- Prześledzisz informacje dotyczące sortowania danych.
- Przeanalizujesz krok po kroku implementację algorytmu sortowania bąbelkowego w języku C++.
- Zaimplementujesz program realizujący algorytm sortowania bąbelkowego.

Animacja

Polecenie 1

Zapoznaj się z animacją, w której omówiono sortowanie bąbelkowe w języku C++.



Film dostępny pod adresem </preview/resource/R1MTHLr42ZLGj>

Źródło: reż. Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Film nawiązujący do treści materiału: Sortowanie bąbelkowe w języku C++.

Plik o rozmiarze 457.00 B w języku polskim

Problem 1

Napisz program sortujący wyniki ankiety dotyczącej problemu tortur w kontekście przestrzegania praw człowieka, wykorzystując sortowanie bąbelkowe (*bubble sort*).

Specyfikacja problemu:

Dane:

- ankietę przeprowadzono w grudniu 2008 roku przez CBOS,
- zadano pytanie: „Jaki jest pana/pani stosunek do tortur?”

- 62% odpowiedziało: „Tortury powinny być zawsze zakazane”, 20% odpowiedziało: „Akceptacja ograniczonego stosowania tortur wobec terrorystów dla ocalenia niewinnych ludzi”, 11% nie miało zdania, 7% odpowiedziało: „Tortury powinny być ogólnie dozwolone”.

Polecenie 2

Porównaj swoje rozwiązanie z przedstawionym w filmie.



Sortowanie tablicy metodą bąbelkową

Algorytm i jego realizacja w języku C++



Film dostępny pod adresem </preview/resource/Rl2cVd3cybTAb>

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Film nawiązujący do treści materiału: Sortowanie tablicy metodą bąbelkową.

Kod programu zaprezentowanego w filmie:

Plik o rozmiarze 548.00 B w języku polskim

Przeczytaj

Przypomnijmy kilka aspektów związanych ze składnią języka C++:

- Tablicę o typie całkowitym (Integer) i wielkości 7 deklarujemy w następujący sposób: `int tab[7]`.
- W celu uzyskania konkretnego elementu tablicy należy użyć notacji `tab[i]`, gdzie `tab` to nazwa tablicy, a `i` to indeks elementu, do którego się odwołujemy.

Polecenie 1

Przeszukaj dostępne źródła i odpowiedz na pytania:

- W jaki sposób zamienia się elementy tablicy miejscami przy założeniu, że nie można wykorzystać funkcji `zamień(ciąg[i], ciąg[i+1])`?
- Czy rozwiązanie zaprezentowane w prezentacji jest w pełni optymalne?
- Jakie usprawnienia można wprowadzić, aby uniknąć niepotrzebnych [iteracji](#) i skrócić działanie algorytmu?

Przykładowa implementacja algorytmu sortowania bąbelkowego w języku C++

Przeanalizujemy przykładową [implementację](#) algorytmu sortowania bąbelkowego. Przedstawiony program posortuje niemalejąco zadany ciąg liczb całkowitych zawartych w tablicy. Implementacja będzie bazować na porównywaniu i ewentualnej zamianie kolejności elementów.

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Alternatywna implementacja algorytmu sortowania bąbelkowego w języku C++

Ponownie stworzymy program sortujący niemalejąco ciąg liczb całkowitych zapisanych w tablicy. Tym razem algorytm zaimplementujemy, bazując na

porównaniu wybranego elementu z elementem go poprzedzającym zamiast porównania z elementem następującym po nim.

Ważne!

Obie przedstawione w tej sekcji implementacje są poprawnymi implementacjami algorytmu sortowania bąbelkowego.

Ciekawostka

Nazwa sortowanie bąbelkowe nawiązuje do faktu, że dane z każdą iteracją są wyrzucane do góry, podobnie jak bąbelki w napojach gazowanych.

Słownik

implementacja

zapis abstrakcyjnego opisu algorytmu jako programu w wybranym języku programowania

iteracja

wielokrotne wykonywanie tych samych operacji (konkretną liczbę powtórzeń lub do spełnienia danego warunku); stosowana najczęściej w pętlach

zagnieżdżenie pętli

wykonywanie pętli wewnątrz innej istniejącej pętli

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Zaprezentowany kod jest implementacją algorytmu sortowania bąbelkowego. Program sortuje elementy tablicy nierosnąco. Zmodyfikuj go tak, aby sortował elementy tablicy niemalejąco.

Przetestuj swój program dla tablicy składającej się z następujących elementów: 7, 8, 5, 6, 3, 2, 1, 9, 4, 0.

Specyfikacja problemu:

Dane:

- liczby – tablica liczb naturalnych
- rozmiar – liczba naturalna; rozmiar tablicy

Wynik:

- liczby – tablica liczb naturalnych posortowana niemalejąco

Ćwiczenie 2



Zaprezentowany kod jest implementacją algorytmu sortowania bąbelkowego. Program do zamiany miejscami elementów tablicy wykorzystuje zmienną pomocniczą *pom*. Zaproponuj inny sposób zamiany elementów miejscami. Pamiętaj o niezbędnych bibliotekach.

Specyfikacja problemu:

Dane:

- *liczby* – tablica liczb naturalnych
- *rozmiar* – rozmiar tablicy; liczba całkowita

Wynik:

- *liczby* – tablica liczb naturalnych posortowana niemalejąco

Ćwiczenie 3



W pewnej kręgielni zapisywano najlepsze dzienne wyniki. Po tygodniu przyszedł czas wyznaczenia rankingu graczy. Użyj sortowania bąbelkowego, aby wydrukować posortowaną rosnąco listę wyników. Użyj wersji zoptymalizowanej algorytmu.

Specyfikacja problemu:

Dane:

- *wyniki* – tablica liczb naturalnych
- *rozmiar* – rozmiar tablicy; liczba całkowita

Wynik:

- *wyniki* – tablica liczb naturalnych

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Sortowanie bąbelkowe w języku C++

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres podstawowy

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy:

c) porządkowania ciągu liczb: przez wstawianie i metodą bąbelkową,

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Prześledzisz informacje dotyczące sortowania danych.

- Przeanalizujesz krok po kroku implementację algorytmu sortowania bąbelkowego w języku C++.
- Zaimplementujesz program realizujący algorytm sortowania bąbelkowego.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Sortowanie bąbelkowe w języku C++”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Animacja”.

Faza wstępna:

1. Nauczyciel wyświetla uczniom temat zajęć oraz cele. Prosi, by na ich podstawie uczniowie sformułowali kryteria sukcesu.

Faza realizacyjna:

1. **Praca z multimediami.** Nauczyciel sprawdza przygotowanie uczniów do zajęć, prosząc wybrane osoby o przedstawienie najważniejszych informacji przedstawionych w sekcji „Animacja”.
2. **Praca z tekstem.** Uczniowie pracując w parach, analizują krok po kroku implementację algorytmu sortowania bąbelkowego przedstawioną w sekcji „Przeczytaj”. Następnie testują omówione rozwiązania na swoich komputerach. Na forum klasy dzielą się swoimi spostrzeżeniami.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenie nr 1, a następnie porównują swoje odpowiedzi z kolegą lub koleżanką.
4. W kolejnym etapie uczniowie dobierają się w pary i wykonują ćwiczenia nr 2 i 3. Następnie konsultują swoje rozwiązania z inną parą uczniów i ustalają jedną wersję odpowiedzi.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się”.

Praca domowa:

1. Uczniowie wykonują problem z sekcji „Animacja”. Ich zadaniem jest napisanie programu sortującego wyniki ankiety dotyczącej problemu tortur w kontekście przestrzegania praw człowieka, wykorzystując sortowanie bąbelkowe.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.

Wskazówki metodyczne:

- Uczniowie mogą wykorzystać treści w sekcjach: „Animacja”, „Przeczytaj”, „Sprawdź się” jako materiał do lekcji powtórkowej.