PIERWSZA ŁAWKA · KURS INFORMATYKA · INFORMATYKA · MATURA PR · ROZSZERZONY

Wyszukiwanie i sortowanie

Wyszukiwanie liniowe i binarne, metoda połowienia przy rozwiązywaniu równań, jednoczesne wyszukiwanie minimum i maksimum, sortowanie bąbelkowe, przez wstawianie, przez scalanie i szybkie (objaśnienie), podciągi o zadanych własnościach, złożoność obliczeniowa. Sześć kart pracy zgodnych z podstawą 2024 i maturą 2027 – z zadaniem przy komputerze na plikach danych (archiwum ZIP), kluczem, zasadami oceniania i numerami wymagań.

Karta	Temat	Zadania
1	**Wyszukiwanie i sortowanie – Teoria krok po kroku** Wszystko, co trzeba wiedzieć, z przykładami	2 zad. 4 pkt
2	**Wyszukiwanie i sortowanie – Rozgrzewka** Proste zastosowania – sprawdź, czy rozumiesz	3 zad. 9 pkt
3	**Wyszukiwanie i sortowanie – Zadania maturalne, część 1** Analiza algorytmów, tabele i zapis algorytmu – jak w arkuszu	4 zad. 13 pkt
4	**Wyszukiwanie i sortowanie – Zadania maturalne, część 2** Trudniejsze zadania i uzasadnienia	4 zad. 15 pkt
5	**Wyszukiwanie i sortowanie – Zadanie z danymi – jak na egzaminie** Praca z plikami danych przy komputerze – wyniki w pliku wyniki1.txt	1 zad. 11 pkt
6	**Wyszukiwanie i sortowanie – Sprawdź się** 30 minut · 13 pkt	5 zad. 13 pkt

JAK KORZYSTAĆ Z ZESTAWU

- Karta 1 – teoria z przykładami. Karty 2–4 – zadania w formie maturalnej: analiza algorytmów, tabele, zapis algorytmu ze specyfikacją (Dane, Wynik) i ograniczeniami z uwagi pod treścią.
- Karta 5 – zadanie przy komputerze: w archiwum ZIP pełne dane i pliki *_przyklad.txt z odpowiedziami kontrolnymi; wyniki w pliku wyniki1.txt z numerami zadań, kod w osobnych plikach. Karta 6 – zestaw kontrolny na czas.
- Ramki „Uwaga na maturze” – błędy, przez które zdający najczęściej tracą punkty (sprawozdania CKE 2023–2026).
- Klucz: rozwiązania wzorcowe (Python; C++ i Java są równoprawne), zasady oceniania i wymagania podstawy (Dz.U. 2024 poz. 1019): I.n, II.n, I+II.n – zakres rozszerzony, P.… – podstawowy. Wyniki dla plików danych policzono programem.

Materiał oryginalny zgodny z podstawą programową informatyki w brzmieniu z 2024 r. (Dz.U. 2024 poz. 1019) i z formułą egzaminu maturalnego z informatyki w 2027 r. Nie zawiera zadań z arkuszy CKE. Do użytku własnego i pracy z własnymi uczniami.

Wyszukiwanie i sortowanie – Teoria krok po kroku

Wszystko, co trzeba wiedzieć, z przykładami

KARTA 1/6

Imię i nazwisko: ..... Data: ..... Wynik: ..... / 4 pkt

PRZYPOMNIJ SOBIE

- **Wyszukiwanie liniowe** przegląda elementy po kolei – O(n) porównań. Działa na dowolnych danych.
- **Wyszukiwanie binarne** (przez połowienie) działa tylko na ciągu *uporządkowanym*: porównujemy x ze środkiem przedziału i odrzucamy połowę. Wykonuje około $\log_2 n$ porównań – dla miliona elementów około 20. Każdy krok musi wykluczać sprawdzony środek ($s + 1$ albo $s - 1$), inaczej pętla może się nie zakończyć.
- **Metoda połowienia dla równań** $f(x) = 0$: gdy f jest ciągła i $f(a) \cdot f(b) < 0$, bierzemy $c = (a + b)/2$ i zostawiamy tę połowę, na której końcach f ma różne znaki. Po k krokach przedział ma długość $(b - a)/2^k$.
- **Jednoczesne minimum i maksimum**: porównujemy elementy parami, mniejszy z pary z minimum, większy z maksimum – około $3n/2$ porównań zamiast $2n - 2$.
- **Sortowanie bąbelkowe**: porównujemy sąsiednie elementy i zamieniamy je, gdy stoją źle; po i-tym przebiegu i największych elementów stoi na końcu. Liczba zamian = liczba *inwersji* (par w złej kolejności). Przebieg bez zamian kończy sortowanie.
- **Sortowanie przez wstawianie**: kolejny element wstawiamy na właściwe miejsce w posortowanym początku, przesuwając większe w prawo. Pesymistycznie $O(n^2)$, dla danych prawie uporządkowanych prawie O(n). Miejsce wstawienia można znaleźć wyszukiwaniem binarnym (mniej porównań, ale tyle samo przesunięć).
- **Sortowanie przez scalanie** (dziel i zwyciężaj): dzielimy ciąg na połowy, sortujemy je rekurencyjnie i scalamy w czasie O(n). Zawsze O(n log n), potrzebuje dodatkowej pamięci. Na maturze wolno użyć funkcji bibliotecznej (sorted, std::sort, Arrays.sort), chyba że polecenie każe zapisać algorytm.
- **Spójne podciągi**: najdłuższy fragment niemalejący znajdziemy w jednym przebiegu (licznik długości zerowany, gdy $A[i] < A[i - 1]$); fragment o największej sumie – też w jednym przebiegu: sumę bieżącą zaczynamy od nowa, gdy przestaje być dodatnia.
- **Sortowanie szybkie** (dziel i zwyciężaj, podstawa I+II.3c – objaśnić i porównać): dzielimy ciąg względem elementu osiowego na elementy mniejsze i większe od niego, a potem sortujemy obie części rekurencyjnie. Średnio O(n log n), pesymistycznie $O(n^2)$ (np. ciąg już uporządkowany, gdy osią jest pierwszy element); nie potrzebuje dodatkowej tablicy. *Dla chętnych (poza podstawą):* sortowanie stabilne zachowuje kolejność elementów o równych kluczach.

Algorytm	Średnio	Pesymistycznie
wyszukiwanie liniowe	O(n)	O(n)
wyszukiwanie binarne	O(log n)	O(log n)
bąbelkowe	$O(n^2)$	$O(n^2)$
przez wstawianie	$O(n^2)$	$O(n^2)$
przez scalanie	O(n log n)	O(n log n)
szybkie	O(n log n)	$O(n^2)$

- **Jak zapisać algorytm, żeby dostać punkty**: pseudokod albo wybrany język (Python, C++, Java) – oba zapisy są równoprawne. Trzymaj się nazw ze *Specyfikacji* (Dane/Wynik), na końcu jawnie ustal wynik (zwróć go albo wypisz), przestrzegaj ograniczeń z *Uwagi* pod treścią i sprawdź algorytm „na sucho" na przykładzie z treści oraz na przypadku brzegowym (np. n = 1, liczba 0).

Podgląd • Pierwsza Ławka

PRZYKŁADY Z WYJAŚNIENIEM

Przykład 1. Ile porównań wykona pesymistycznie wyszukiwanie binarne w ciągu 1000 elementów?
Przedział maleje: 1000 → 500 → 250 → 125 → 62 → 31 → 15 → 7 → 3 → 1, czyli najwyżej 10 porównań ($2^{10} = 1024 > 1000$).

Przykład 2. Dlaczego wyszukiwania binarnego nie można użyć dla ciągu nieuporządkowanego?
Algorytm odrzuca połowę na podstawie porównania ze środkiem. W ciągu nieuporządkowanym x może leżeć po obu stronach środka.

Przykład 3. Wykonaj jeden przebieg sortowania bąbelkowego ciągu 5, 2, 9, 1.
(5, 2) zamiana → 2, 5, 9, 1; (5, 9) bez zmian; (9, 1) zamiana → 2, 5, 1, 9. Największy element jest na końcu.

Przykład 4. Ile kroków metody połowienia potrzeba, żeby przedział [0, 1] zawęzić do długości mniejszej niż 0,01?
Po k krokach długość to $1/2^k$; $1/2^6 = 1/64 > 0{,}01$, a $1/2^7 = 1/128 < 0{,}01$ – potrzeba 7 kroków.

ZADANIA

Zadanie 1. W uporządkowanym ciągu 2, 5, 8, 12, 16, 23, 38, 56, 72, 91 (indeksy od 0 do 9) szukamy 23 wyszukiwaniem binarnym, biorąc środek s = (lo + hi) div 2. Wypisz kolejne elementy porównywane z 23. ___ / 3 pkt

Zadanie 2. Ile porównań wykona pesymistycznie wyszukiwanie binarne w uporządkowanym ciągu miliona elementów? ___ / 1 pkt

A. około 10
B. około 20
C. około 1000
D. około 500 000

Podgląd • Pierwsza Ławka

Wyszukiwanie i sortowanie – Rozgrzewka

KARTA 2/6

Proste zastosowania – sprawdź, czy rozumiesz

Imię i nazwisko: ..... Data: ..... Wynik: ..... / 9 pkt

Zadanie 1. Posortuj rosnąco ciąg 5, 1, 4, 2, 8 algorytmem bąbelkowym. Zapisz ciąg po każdym przebiegu i podaj łączną liczbę zamian. ___ / 3 pkt

Zadanie 2. Oceń prawdziwość zdań. ___ / 4 pkt

1. Sortowanie przez scalanie ma złożoność O(n log n) także w przypadku pesymistycznym.	**P**	**F**
2. Sortowanie przez wstawianie ciągu n elementów uporządkowanego już rosnąco wykonuje n − 1 porównań elementów.	**P**	**F**
3. Wyszukiwanie binarne działa poprawnie także w ciągu nieuporządkowanym.	**P**	**F**
4. Sortowanie bąbelkowe ma pesymistyczną złożoność O(n log n).	**P**	**F**

Zadanie 3. Ile porównań elementów wykona dla n = 100 a) wyszukiwanie minimum i maksimum dwoma osobnymi przebiegami, b) jednoczesne wyszukiwanie parami? ___ / 2 pkt

Podgląd • Pierwsza Ławka

Wyszukiwanie i sortowanie – Zadanie z danymi – jak na egzaminie

Praca z plikami danych przy komputerze – wyniki w pliku wyniki1.txt

KARTA 5/6

Imię i nazwisko: ..... Data: ..... Wynik: ..... / 11 pkt

Zadanie 1. W pliku zmiany.txt zapisano 2000 liczb całkowitych z przedziału [−60, 60] – dzienne zmiany poziomu wody w zbiorniku (w cm), każdą w osobnym wierszu; wiersz 1 to dzień 1. W pliku zapytania.txt jest 200 liczb całkowitych, po jednej w wierszu. Napisz program (lub kilka programów), który da odpowiedzi do poniższych zadań. Odpowiedzi zapisz w pliku wyniki1.txt, poprzedzając każdą numerem zadania. ___ / 11 pkt

Pierwsze wiersze pliku zmiany.txt

```
-30
20
-36
14
0
54
…
```

Do dyspozycji masz także pliki **zmiany_przyklad.txt** (40 liczb) i **zapytania_przyklad.txt** (10 liczb) – odpowiedzi dla nich podano w treściach zadań. Twój program musi ostatecznie działać na pełnych danych (**zmiany.txt**, **zapytania.txt**).

Do oceny oddajesz: plik **wyniki1.txt** z odpowiedziami do zadań 1.1–1.4 (każda poprzedzona numerem zadania) oraz pliki z kodem źródłowym, np. **zad1_1.py**, **zad1_2.py**… (lub .cpp, .java). Brak plików z kodem oznacza brak rozwiązania.

Dane przygotowane na potrzeby kursu (archiwum ZIP dołączone do opracowania, katalog DANE)

UWAGA NA MATURZE Numer wiersza to indeks w liście + 1 – przy odpowiedziach z numerami wierszy to najczęstszy błąd „o jeden”. Plik przykładowy służy właśnie do jego wyłapania.

1.1. Podaj najmniejszą i największą liczbę zapisaną w pliku zmiany.txt. Odpowiedź dla danych przykładowych: -58 54 ___ / 2 pkt

1.2. Znajdź najdłuższy spójny fragment pliku zmiany.txt, w którym liczby tworzą ciąg niemalejący (każda kolejna jest większa lub równa poprzedniej). Podaj jego długość i numer wiersza, w którym się zaczyna. Jest jeden taki fragment. Odpowiedź dla danych przykładowych: 9 10 ___ / 3 pkt

UWAGA NA MATURZE „Niemalejący” to $A[i] \geq A[i-1]$ – z równością. Warunek ścisły (>) daje ciąg rosnący i inny wynik.

Podgląd • Pierwsza Ławka

Wyszukiwanie i sortowanie – Sprawdź się

30 minut · 13 pkt

KARTA 6/6

Imię i nazwisko: ..... Data: ..... Wynik: ..... / 13 pkt

Zadanie 1. Oceń prawdziwość zdań. ___ / 4 pkt

1. Wyszukiwanie liniowe działa także w ciągu nieuporządkowanym.	**P**	**F**
2. Liczba zamian w sortowaniu bąbelkowym jest równa liczbie inwersji w ciągu.	**P**	**F**
3. Sortowanie przez wstawianie ma pesymistyczną złożoność $O(n \log n)$.	**P**	**F**
4. Metodą połowienia można przybliżać pierwiastek równania $f(x) = 0$, gdy f jest ciągła i $f(a) \cdot f(b) < 0$.	**P**	**F**

Zadanie 2. Ile zamian wykona sortowanie bąbelkowe ciągu 3, 1, 2, 5, 4? Wypisz pary tworzące inwersje. ___ / 2 pkt

Zadanie 3. Ile porównań wykona scalanie ciągów uporządkowanych 1, 4, 9 oraz 2, 3, 10? Zapisz wynik scalania. ___ / 2 pkt

Zadanie 4. Dla jakich danych sortowanie przez wstawianie wykona wyraźnie mniej porównań niż sortowanie przez scalanie? Odpowiedź uzasadnij, porównując liczby porównań dla dużego n. ___ / 2 pkt

Zadanie 5. Oszacuj liczbę operacji dla $n = 10^6$ dla algorytmów o złożoności $O(n^2)$ i $O(n \log_2 n)$. Który z nich wykona się w sekundę na komputerze wykonującym 10^8 operacji na sekundę? ___ / 3 pkt

Podgląd • Pierwsza Ławka

CO JESZCZE JEST W ŚRODKU

Wyszukiwanie i sortowanie – opracowanie maturalne z informatyki (6 kart pracy z kluczem)

Podgląd pokazał 6 z 17 stron. Pełna wersja: 17 stron A4, PDF do druku i DOCX do edycji.

- Karta 1: Teoria krok po kroku – 2 zadania, 4 pkt
- Karta 2: Rozgrzewka – 3 zadania, 9 pkt
- Karta 3: Zadania maturalne, część 1 – 4 zadania, 13 pkt
- Karta 4: Zadania maturalne, część 2 – 4 zadania, 15 pkt
- Karta 5: Zadanie z danymi – jak na egzaminie – 1 zadanie, 11 pkt
- Karta 6: Sprawdź się – 5 zadań, 13 pkt Razem 17 stron A4 w PDF do druku oraz ten sam materiał w DOCX do edycji w Wordzie, a do tego archiwum ZIP z plikami danych do zadania z karty

Kup pełną wersję – plik od razu na e-mail
pierwszalawka.pl/material/matura-informatyka-wyszukiwanie-i-sortowanie-opracowanie