SPRAWDZIAN · INFORMATYKA

Sprawdzian z algorytmiki **Grupa A**

Informatyka: algorytmy klasyczne, rekurencja, techniki projektowania i złożoność

Imię i nazwisko: _____

Klasa: _____

Data: _____

Punkty: _____ / 28

10 zadań, 28 pkt. Czytaj uważnie polecenia. W zadaniach otwartych zapisz obliczenia.

Zadanie 1. (0–1 pkt)

Algorytm sortowania przez scalanie (mergesort) tablicy n elementów ma złożoność pesymistyczną:

(A) $O(n^2)$
(B) $O(n \log n)$
(C) $O(n)$
(D) $O(\log n)$

Zaznacz jedną poprawną odpowiedź.

Zadanie 2. (0–1 pkt)

Problem plecakowy 0-1 (knapsack) dla n przedmiotów i pojemności W ma złożoność w podejściu programowania dynamicznego:

(A) $O(2^n)$
(B) $O(n!)$
(C) $O(n^2)$
(D) $O(n \cdot W)$

Zaznacz jedną poprawną odpowiedź.

Zadanie 3. (0–3 pkt)

Oceń prawdziwość stwierdzeń dotyczących algorytmów sortowania:

1.	Algorytm QuickSort z pivotem zawsze wybieranym jako pierwszy element tablicy dla danych już posortowanych działa w czasie O(n^2).	**P**	**F**
2.	Merge Sort jest algorytmem stabilnym o złożoności pesymistycznej O(n log n).	**P**	**F**
3.	Algorytm sortowania jest stabilny, jeśli elementy o równych kluczach zachowują swój pierwotny wzajemny porządek po sortowaniu.	**P**	**F**

Przy każdym zdaniu zakreśl P (prawda) albo F (fałsz).

Zadanie 4. (0–3 pkt)

Oceń prawdziwość stwierdzeń o algorytmach sortowania:

1.	Sortowanie pozycyjne (radix sort) ma złożoność $O(n \cdot k)$, gdzie k to liczba cyfr.	**P**	**F**
2.	Żaden algorytm sortowania przez porównania nie może działać szybciej niż $O(n \log n)$ w pesymistycznym przypadku.	**P**	**F**
3.	Sortowanie przez kopcowanie (heapsort) wymaga dodatkowej pamięci $O(n)$.	**P**	**F**

Przy każdym zdaniu zakreśl P (prawda) albo F (fałsz).

Zadanie 5. (0–4 pkt)

Dopasuj technikę algorytmiczną do problemu, który rozwiązuje najefektywniej.

1.	Programowanie zachłanne	...	A. Sortowanie przez scalanie
2.	Przeszukiwanie z powrotami (backtracking)	...	B. Generowanie wszystkich permutacji
3.	Dziel i zwyciężaj (divide & conquer)	...	C. Problem wydawania reszty (minimum monet)
4.	Programowanie dynamiczne	...	D. Minimalne drzewo rozpinające (Kruskal)

Przy każdym numerze wpisz literę pasującego elementu.

Zadanie 6. (0–4 pkt)

Dopasuj algorytm do jego pesymistycznej złożoności czasowej:

1. Bubble sort	...	A. $O(n^2)$
2. Merge sort	...	B. $O(\log n)$
3. Wyszukiwanie binarne	...	C. $O(n)$
4. Wyszukiwanie liniowe	...	D. $O(n \log n)$

Przy każdym numerze wpisz literę pasującego elementu.

Zadanie 7. (0–3 pkt)

Wyjaśnij, czym jest notacja Big O. Podaj przykłady algorytmów o złożonościach $O(1)$, $O(\log n)$, $O(n)$ i $O(n^2)$.

Zadanie 8. (0–3 pkt)

Wyjaśnij, czym jest algorytm zachłanny (greedy). Podaj jeden przykład problemu, do którego rozwiązania stosuje się podejście zachłanne, i wyjaśnij, na czym polega podejmowanie decyzji w tym podejściu.

Zadanie 9. (0–3 pkt)

Wyjaśnij zasadę działania algorytmu wyszukiwania binarnego. Podaj jego złożoność czasową w najgorszym przypadku i warunek wstępny, który musi spełniać zbiór danych.

Zadanie 10. (0–3 pkt)

Wyjaśnij, na czym polega technika programowania dynamicznego i czym różni się od zwykłej rekurencji wspomaganej memoizacją pod względem sposobu budowania rozwiązania. Podaj jeden przykład problemu, do którego rozwiązania typowo stosuje się programowanie dynamiczne.

Rozwiąż ten sprawdzian online, z wyjaśnieniem każdej odpowiedzi: matury-online.pl/sprawdzian/informatyka/algorytmika