

Egzamin licencjacki/inżynierski

17 lutego 2025

Informacja dla zdających egzamin na kierunku informatyka

Z sześciu poniższych zestawów zadań (Matematyka I, Matematyka II, Metody programowania, Matematyka dyskretna, Algorytmy i struktury danych, Metody numeryczne) należy wybrać i przedstawić na osobnych kartkach rozwiązania trzech zestawów.

Informacja dla zdających egzamin na kierunku ISIM

Z sześciu poniższych zestawów zadań (Matematyka I, Metody programowania, Matematyka dyskretna, Algorytmy i struktury danych, Metody numeryczne, Języki formalne i złożoność obliczeniowa) należy wybrać i przedstawić na osobnych kartkach rozwiązania trzech zestawów.

Informacja dla wszystkich zdających

Za brakujące (do trzech) zestawy zostanie wystawiona ocena niedostateczna z urzędu. Egzamin uważa się za zaliczony, jeśli student rozwiąże z oceną dostateczną co najmniej 2 zestawy. Wtedy ocena z egzaminu jest średnią arytmetyczną ocen z trzech wybranych zestawów. Na rozwiązanie przeznaczona jest czas $3 \times 40 + 30 = 150$ minut. Po wyjściu z sali egzaminacyjnej w czasie egzaminu nie ma możliwości powrotu do tej sali i kontynuowania pisania egzaminu.

Matematyka I — Logika dla informatyków

Mówimy, że formuła φ rachunku zdań nad zbiorem zmiennych zdaniowych V *istotnie zależy od zmiennej* p jeśli istnieją takie różne wartościowania σ i σ' , że $\hat{\sigma}(\varphi) \neq \hat{\sigma}'(\varphi)$ oraz $\sigma(v) = \sigma'(v)$ dla wszystkich zmiennych zdaniowych v z wyjątkiem p . Udowodnij, że jeśli formuła φ nie zależy istotnie od zmiennej p , to istnieje równoważna jej formuła, w której zmienna p nie występuje.

Matematyka II — Algebra

Zadanie 1. (4 punkty)

W przestrzeni Π_2 wielomianów stopnia ≤ 2 definiujemy iloczyn skalarny

$$\langle f, g \rangle = \int_0^2 f(x) g(x) dx$$

Proszę wykonać ortogonalizację bazy potęgowej tej przestrzeni.

Zadanie 2. (5 punktów)

Niech L będzie nieosobliwą macierzą trójkątną dolną. Wykazać, że L^{-1} też jest macierzą trójkątną dolną.

Zadanie 3. (6 punktów)

Ile rozwiązań ma poniższy układ w zależności od parametru λ ? Układ jest nad \mathbb{Z}_7 .

$$\begin{bmatrix} \lambda & \lambda^2 \\ 1 & \lambda^2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ \lambda \end{bmatrix}.$$

Progi punktowe: 5, 7, 9, 11, 13 punktów.

Matematyka dyskretna

Założmy, że wszystkie wierzchołki grafu spójnego G mają parzyste stopnie. Wykaż, że graf G nie ma mostów.

Metody Programowania

Poniższe zadania należy rozwiązać używając języka Plait lub OCaml.

Zadanie 1 Zdefiniuj funkcję `merge` scalającą dwie posortowane listy liczb do pojedynczej posortowanej listy liczb. Wszystkie elementy należące do list wejściowych (włącznie z powtórzeniami) powinny znaleźć się w liście wyjściowej.

Zadanie 2 Zdefiniuj funkcję-predykat sprawdzającą, czy lista będąca jej argumentem jest posortowana. Następnie udowodnij przez indukcję, że jeśli argumenty funkcji `merge` są posortowanymi listami, to wynik tej funkcji również jest posortowany.

Zadanie 3 Przyjmijmy, że funkcja `split`, przyjmująca jako argument listę liczb i zwracająca parę list liczb, jest już zdefiniowana. Zdefiniuj funkcję `mergesort` implementującą algorytm sortowania przez scalanie przy użyciu funkcji `split` i `merge`.

Zadanie 4 Udowodnij przez indukcję, że funkcja `mergesort` zwraca wyłącznie listy posortowane.

Metody numeryczne

Za rozwiązanie zadań można otrzymać łącznie 12 punktów. Otrzymanie 4 pkt. gwarantuje ocenę dostateczną, próg dla dst+ to 5.5 pkt., dla db – 7 pkt., dla db+ 8.5 pkt., a dla bdb – 10 pkt.

1. **4 punkty** Niech dana będzie funkcja $f(x) := 8100 \frac{\sqrt{x^{13} + 4} - 2}{x^{13}}$. W trybie podwójnej precyzji wartość $\text{fl}(f(0.001)) = 0.0$. **Uzasadnij**, że wynik ten nie odpowiada rzeczywistej wartości $f(0.001)$. **Wytlumacz** dlaczego tak się dzieje i **zaproponuj** sposób obliczenia wyniku dokładniejszego.
2. **4 punkty** Niech będzie $\mathbf{x} := [x_0, x_1, \dots, x_n]$ ($x_0 < x_1 < \dots < x_n$), $\mathbf{y} := [y_0, y_1, \dots, y_n]$ oraz $\mathbf{z} := [z_0, z_1, \dots, z_m]$. Niech s_n oznacza NIFS3 spełniającą warunki $s_n(x_k) = y_k$ ($0 \leq k \leq n$). W języku PW0++ procedura `NSpline3(x,y,z)` wyznacza wektor $\mathbf{Z} := [s_n(z_0), s_n(z_1), \dots, s_n(z_m)]$, z tym, że **musi być** $m < 2n$. Załóżmy, że wartości pewnej funkcji $f \in C^1[x_0, x_1]$ znane są **jedynie** w punktach $x_0 < x_1 < \dots < x_{100}$. Wiadomo, że **pochodna** NIFS3 odpowiadającej danym $(x_k, f(x_k))$ ($0 \leq k \leq 100$) bardzo dobrze przybliża funkcję f' . Wywołując procedurę `NSpline3` **tylko raz**, opracuj algorytm numerycznego wyznaczania przybliżonych wartości wszystkich miejsc zerowych f' leżących w przedziale $[x_0, x_{100}]$.
3. **4 punkty** Zaproponuj i opisz **szczegółowo** algorytm obliczania wyznacznika macierzy stopnia n . Wyznacz jego złożoność obliczeniową i pamięciową.

Uwaga. Metody o złożoności obliczeniowej powyżej $O(n^3)$ nie wchodzi w grę.

Algorytmy i struktury danych

Zadanie 1: minimalny łączny czas oczekiwania na obsługę (4 punkty)

Do pewnego biura w urzędzie miejskim przyszło n osób. Każda osoba chce załatwić swoją typową sprawę, tak więc czas obsługi i -tego petenta jest znany i wynosi t_i minut. Przykładowo, jeśli petenci są ustawieni w porządku rosnących numerów, to i -ta osoba musi czekać na obsługę $\sum_{j=1}^i t_j$ minut. Należy tak ustawić petentów w kolekę, aby zminimalizować łączny czas oczekiwania na obsługę:

$$T = \sum_{i=1}^n (\text{czas oczekiwania } i\text{-tej osoby})$$

Opracuj efektywny algorytm rozwiązujący to zadanie. Precyzyjnie opisz ideę rozwiązania a potem zapisz ją w pseudokodzie. Uzasadnij poprawność opisanej metody i oszacuj jej złożoność obliczeniową.

Zadanie 2: usuwanie elementów ze słownika statycznego (5 punktów)

Słownik statyczny to struktura danych, którą najpierw wypełniamy określonymi wartościami a potem tylko sprawdzamy czy podane wartości znajdują się w takiej strukturze. Do implementacji słownika statycznego często wykorzystujemy tablice z haszowaniem.

W kontekście tablic z haszowaniem odpowiedz na następujące pytania:

- Co to jest funkcja haszująca i jaką spełnia rolę.
- Na czym polega metoda adresowania otwartego w tablicach z haszowaniem?
- Jakie znasz sposoby rozwiązywania konfliktów w przypadku adresowania otwartego (konflikty mogą się pojawić podczas wstawiania do niej nowych elementów)?

Rozważmy tablicę z haszowaniem z zastosowaniem adresowania otwartego. Do ciągu zapytań $search(x)$ o istnienie zadanej wartości x w tablicy dokładamy jeszcze operację $delete(x)$, która ma usunąć element x z tablicy. Opisz, jak należy zaimplementować usuwanie niepotrzebnych elementów z tablicy z haszowaniem, przy zastosowaniu adresowania otwartego. Jak należy wówczas zmodyfikować procedurę wyszukiwania, aby działała poprawnie nawet po usunięciu jakiegoś elementu ze ścieżki poszukiwań? Przeanalizuj czasy działania obu metod $search(x)$ i $delete(x)$.