

Egzamin licencjacki/inżynierski

21 lutego 2022

Informacja dla zdających

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.

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

Udowodnij, że dla każdego $n > 2$ istnieje taki n -gumentowy spójnik logiczny \otimes , że $\{\otimes\}$ jest zupełnym zbiorem spójników.

Matematyka II — Algebra

Zadanie 1. (4 punkty)

Przekształcenie liniowe $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ jest bijekcją. Wykazać, że przekształcenie odwrotne φ^{-1} jest (też) przekształceniem liniowym.

Zadanie 2. (4 punkty)

Niech $A, B \in \mathbb{R}^{n \times n}$. $A = [c_1 : c_2 : \dots : c_n]$ (czyli c_i to i -ta kolumna macierzy A) oraz $B = (b_{ij})$. W jaki sposób elementy b_{ij} opisują operacje na kolumnach macierzy A ?

Zadanie 3. (6 punktów)

$$D_n = \begin{vmatrix} a_1 & b_2 & & & & \\ b_2 & a_2 & b_3 & & & \\ & b_3 & a_3 & b_4 & & \\ & & \ddots & \ddots & \ddots & \\ & & & b_{n-1} & a_{n-1} & b_n \\ & & & & b_n & a_n \end{vmatrix}.$$

Udowodnić, że $D_k = a_k D_{k-1} - b_k^2 D_{k-2}$ ($k = 3, 4, \dots$).

Progi punktowe: 4, 6, 8, 10, 12 punktów.

Metody Programowania

Poniższe zadania należy rozwiązać używając języka funkcyjnego. Do wyboru: Racket, OCaml i Haskell.

Zadanie 1 Zdefiniuj dwa typy danych: jeden do reprezentowania list parzystej długości, a drugi do reprezentowania list nieparzystej długości. Możesz ustalić typ elementów listy lub uczynić go parametrem swoich definicji.

Zadanie 2 Dla swoich definicji z zadania 1, napisz funkcję odwracającą listę długości parzystej, a także funkcję odwracającą listę długości nieparzystej.

Zadanie 3 Zaproponuj odpowiednie struktury danych, które wykorzystasz do napisania funkcji, która przyjmie jako argument słownik (zawierający wiedzę o odmianie czasowników w języku angielskim), osobę (pierwszą, drugą lub trzecią), liczbę (pojedynczą lub mnogą) oraz czasownik w bezokoliczniku (np. *be*) i zwróci jego poprawną formę w zadanej osobie zadanej liczby, o ile zadany czasownik znajduje się w słowniku. Napisz tę funkcję. Nie musisz implementować ani przejmować się wydajnością operacji wyszukiwania czasownika w słowniku. Należy natomiast określić jej sygnaturę zakładając, że jej jedynym argumentem jest czasownik w bezokoliczniku.

Uwaga! Jakość kodu, a zwłaszcza struktura definicji, ma znaczenie i będzie punktowana.

Matematyka dyskretna

Niech *cięcie* w grafie to zbiór wszystkich krawędzi łączących pewne zbiory $V' \subset V$ i $V \setminus V'$. Odpowiedz na następujące pytania i uzasadnij te odpowiedzi.

- Czy w grafie o wszystkich wierzchołkach stopnia parzystego wszystkie cięcia mają moc parzystą?
- Czy w grafie o wszystkich wierzchołkach stopnia nieparzystego wszystkie cięcia mają moc nieparzystą?

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 dane będzie $n \in \mathbb{N}$. W języku **PW0++** procedura **Power2Chebyshev**(n) znajduje taki wektor liczb rzeczywistych $[a_0^{(n)}, a_1^{(n)}, \dots, a_n^{(n)}]$, że

$$x^n = \sum_{k=0}^n a_k^{(n)} \cdot T_k(x),$$

gdzie T_k oznacza k -ty wielomian Czebyszewa. Krótko mówiąc: procedura ta wyznacza współczynniki postaci Czebyszewa jednomianu x^n . **Niestety ma ona pewną wadę**, a mianowicie — stopień n nie może być większy niż 2021. Wykorzystując procedurę **Power2Chebyshev**, zaproponuj *efektywną* metodę znajdowania postaci Czebyszewa jednomianu x^{2023} , tj. obliczania współczynników $a_k^{(2023)}$ ($0 \leq k \leq 2023$).

2. **4 punkty** Opisz wybraną metodę iteracyjną przybliżonego rozwiązywania równania nieliniowego (przedstaw ideę i interpretację geometryczną, podaj potrzebne założenia, sformułuj warunek **STOPu**, określ rząd zbieżności, ...).
3. **4 punkty** Załóżmy, że dane są: funkcja ciągła f , liczby $a < b$ oraz parami różne węzły x_0, x_1, \dots, x_n . Niech $Q_n(f)$ będzie kwadraturą interpolacyjną z węzłami x_0, x_1, \dots, x_n przybliżającą wartość całki

$$I(f) := \int_a^b f(x) dx.$$

Jak wiadomo, współczynniki A_k ($0 \leq k \leq n$) kwadratury Q_n ,

$$Q_n(f) := \sum_{k=0}^n A_k f(x_k),$$

wyrażają się wzorem:

$$A_k = \int_a^b \left(\prod_{\substack{i=0 \\ i \neq k}}^n \frac{x - x_i}{x_k - x_i} \right) dx \quad (k = 0, 1, \dots, n).$$

Podaj **efektywny algorytm** obliczania wartości współczynników A_0, A_1, \dots, A_n i określ jego złożoność.

Algorytmy i struktury danych

Za rozwiązanie obydwu zadań z tej części można otrzymać w sumie do 9 punktów. Skala ocen: poniżej 3 punktów — ocena niedostateczna (egzamin niezdany), 3 punkty dają ocenę dostateczną, 4 — dostateczną z plusem, 5 — dobrą, 6 — dobrą z plusem, 7 albo więcej punktów daje ocenę bardzo dobrą.

Zadanie 1: podciąg palindromiczny (4 punkty)

Dany jest n -literowy ciąg $X = (x_1, x_2, \dots, x_n)$. Należy wyznaczyć najdłuższy podciąg w tym ciągu, który jest palindromem (palindrom to słowo brzmiące tak samo czytane od strony lewej do prawej i od prawej do lewej). Zaprojektuj efektywny algorytm rozwiązujący to zadanie. Opisz ideę algorytmu a potem zapisz go w pseudokodzie (wraz z niezbędnymi komentarzami). Uzasadnij, że opisany algorytm działa poprawnie. Oszacuj też złożoność obliczeniową (czasową i pamięciową) przedstawionego rozwiązania.

Wskazówka: jeśli pierwsza i ostatnia litera w tym ciągu będzie taka sama, to czy litery te wejdą w skład optymalnego rozwiązania?

Przykład: Dla ciągu (a, b, c, a, b, a, c) rozwiązaniem może być 5-literowy podciąg $abcba$, $ababa$ albo $cabac$.

Zadanie 2: ciąg elementów z operacjami *insert* oraz *remove* (5 punktów)

Niech $S = (s_0, s_1, \dots, s_{n-1})$ będzie n -elementowym ciągiem elementów. Na ciągu takim chcemy wykonywać następujące operacje:

1. $S = \text{new}()$: $S \leftarrow ()$ utworzenie pustego ciągu;
2. $x = S.\text{select}(i)$: odczytanie i -tego elementu s_i w ciągu S (dla $0 \leq i \leq n-1$);
3. $S.\text{insert}(i, x)$: $S \leftarrow (s_0, \dots, s_{i-1}, x, s_i, \dots, s_{n-1})$ wstawienie nowego elementu x do ciągu S na pozycję i -tą (dla $0 \leq i \leq n$); elementy znajdujące się dotychczas na pozycjach od i do $n-1$ będą przesunięte o jedną pozycję wyżej;
4. $S.\text{remove}(i)$: $S \leftarrow (s_0, \dots, s_{i-1}, s_{i+1}, \dots, s_{n-1})$ usunięcie elementu s_i z pozycji i -tej w ciągu S (dla $0 \leq i \leq n-1$); elementy znajdujące się dotychczas na pozycjach od $i+1$ do $n-1$ będą przesunięte o jedną pozycję niżej.

Zaprojektuj taką strukturę danych dla ciągu S , która umożliwi wykonywanie wymienionych operacji w czasie logarytmicznym $O(\log n)$, gdzie $n = |S|$ (liczba elementów w ciągu S). Wykorzystaj jakąś znaną strukturę danych, która efektywnie realizuje operacje wstawiania i usuwania elementów. Opisz jak zaadoptować tę strukturę na potrzeby zadania. Szczegółowo opisz działanie funkcji *select()*, *insert()* i *remove* (może być pseudokod z komentarzami). Przeanalizuj złożoność czasową wszystkich operacji na tej strukturze.