Egzamin licencjacki/inżynierski

13 lutego 2023

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 przeznacza się czas 3x40+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

Na wykładzie z *Logiki dla informatyków* poznaliśmy system dedukcji naturalnej dla rachunku zdań. Wiemy (i możemy z tego skorzystać w rozwiązaniu tego zadania), że jest on *poprawny*, czyli każda formuła, którą można w nim udowodnić jest tautologią rachunku zdań oraz *zupełny*, czyli można w nim udowodnić każdą tautologię rachunku zdań.

(a) Rozważmy modyfikację systemu dedukcji naturalnej, w którym reguła wprowadzania koniunkcji ($\wedge$i) została zastąpiona regułą

$$\frac{\alpha \quad \alpha \Rightarrow \beta}{\alpha \wedge \beta}\ (\wedge\mathrm{i}')$$

Czy tak uzyskany system jest poprawny? A czy jest zupełny?

(b) Rozważmy modyfikację systemu dedukcji naturalnej, w którym reguła wprowadzania alternatywy ($\vee$i) została zastąpiona regułą

$$\frac{\beta \Rightarrow \alpha \quad \alpha \Rightarrow \beta}{\alpha \vee \beta}\ (\vee\mathrm{i}')$$

Czy tak uzyskany system jest poprawny? A czy jest zupełny?

Poniżej przypominamy komplet reguł dowodzenia systemu dedukcji naturalnej dla rachunku zdań:

$$\frac{\alpha \quad \beta}{\alpha \wedge \beta}\ (\wedge\mathrm{i}) \qquad \frac{\alpha}{\alpha \vee \beta}\ (\vee\mathrm{i}_1) \qquad \frac{\beta}{\alpha \vee \beta}\ (\vee\mathrm{i}_2) \qquad \frac{}{\top}\ (\top\mathrm{i}) \qquad \frac{\boxed{\begin{array}{c}\alpha\\ \vdots\\ \beta\end{array}}}{\alpha \Rightarrow \beta}\ (\Rightarrow \mathrm{i}) \qquad \frac{\boxed{\begin{array}{c}\alpha\\ \vdots\\ \bot\end{array}}}{\neg\alpha}\ (\neg\mathrm{i}) \qquad \frac{\neg\neg\alpha}{\alpha}\ (\neg\neg\mathrm{e})$$

$$\frac{\alpha \wedge \beta}{\alpha}\ (\wedge\mathrm{e}_1) \qquad \frac{\alpha \wedge \beta}{\beta}\ (\wedge\mathrm{e}_2) \qquad \frac{\alpha \vee \beta \quad \boxed{\begin{array}{c}\alpha\\ \vdots\\ \gamma\end{array}} \quad \boxed{\begin{array}{c}\beta\\ \vdots\\ \gamma\end{array}}}{\gamma}\ (\vee\mathrm{e}) \qquad \frac{\alpha \quad \alpha \Rightarrow \beta}{\beta}\ (\Rightarrow \mathrm{e}) \qquad \frac{\alpha \quad \neg\alpha}{\bot}\ (\neg\mathrm{e}) \qquad \frac{\bot}{\alpha}\ (\bot\mathrm{e})$$

Matematyka II — Algebra

Zadanie 1. (5 punktów)
Znaleźć wielomian (możliwie niskiego stopnia) który w punktach 1, 2, 3 przyjmuje wartości 1, 0, 5. UWAGA: chodzi o wielomian nad ciałem $\mathbb{Z}_7$.

Zadanie 2. (5 punktów)
Niech $A, B \in \mathbb{R}^{n \times n}$. Symbolem a_i oznaczamy i-ty wiersz macierzy A. Rozważamy iloczyn $C = B \cdot A$. W jaki sposób elementy b_{ij} opisują operacje na wierszach macierzy A?

Zadanie 3. (4 punkty)

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

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

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

Metody Programowania

Poniższe zadania należy rozwiązać używając języka Racket.

Zadanie 1 Zaimplementuj jednoargumentową procedurę `sorted?`, która rozstrzyga czy dana lista wartości numerycznych jest uporządkowana niemalejąco.

Zadanie 2 Zaimplementuj procedurę `insert-sort`, która sortuje daną listę wartości numerycznych niemalejąco, zgodnie z algorytmem:

- dla listy pustej, zwróć listę pustą;
- dla listy niepustej, wstaw głowę listy na odpowiednią pozycję do posortowanego ogona listy.

Zadanie 3 Udowodnij formalnie, że Twoja procedura `insert-sort` spełnia warunek:

$$\texttt{(sorted? (insert-sort } xs\texttt{))} \equiv \texttt{\#t}$$

dla wszystkich list wartości numerycznych xs.

Matematyka dyskretna

Niech G_1 i G_2 są grafami. Zbiorem wierchołków grafu $G_1 \times G_2$ jest zbiór par (v_1, v_2), gdzie $v_1 \in V(G_1)$ i $v_2 \in V(G_2)$. Wierchołki (v_1, v_2) i (u_1, u_2) są sąsiednie wtedy i tylko wtedy gdy zachodzi jeden z warunków:

- $v_1 = u_1$ i $\{v_2, u_2\} \in E(G_2)$ lub
- $\{v_1, u_1\} \in E(G_1)$ i $v_2 = u_2$.

Pokaż, że

1. jeżeli G_1 i G_2 są r-regularne, to $G_1 \times G_2$ jest regularny (którego stopnia?)
2. jeżeli G_1 i G_2 są dwudzielne, to $G_1 \times G_2$ jest dwudzielny

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. $\boxed{\textbf{4 punkty}}$ **Podaj** definicję naturalnej interpolacyjnej funkcji sklejanej trzeciego stopnia (*w skrócie*: NIFS3). **Znajdź** NIFS3 dla danych

x_k	-2022	-4	-2	0	1	3	2022
y_k	8043	1989	1983	1977	1974	1968	-4089

Uwaga. Rozwiązanie drugiej części zadania nie wymaga wielu obliczeń.

2. $\boxed{\textbf{4 punkty}}$ **Przedstaw szczegółowo** dowolną metodę całkowania numerycznego.

3. $\boxed{\textbf{4 punkty}}$ **Opisz** metodę faktoryzacji rozwiązywania układów równań liniowych i **podaj** jej złożoność czasową i pamięciową. **Znajdź** rozkład LU macierzy

$$A := \begin{bmatrix} 1 & 2 & -1 & 2 \\ -2 & -5 & 3 & -4 \\ 4 & 12 & -10 & 10 \\ -2 & -6 & 8 & -6 \end{bmatrix},$$

a następnie wykorzystaj otrzymany rozkład do **rozwiązania metodą faktoryzacji** (inne metody nie wchodzą w grę!) układu równań $Ax = b$, gdzie $b := [4, -6, 8, 0]^T$.

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: minimalna liczba peronów na stacji (4 punkty)

Stacja kolejowa w pewnym dużym mieście obsługuje tysiące przyjeżdżających i odjeżdżających pociągów dziennie. Każdy pociąg przyjeżdża na stację o określonej godzinie, zatrzymuje się na kilka minut (albo dłużej) na jednym z licznych peronów, po czym o określonej godzinie odjeżdża dalej (wszystko działa punktualnie jak w Japonii).

Dyrektor stacji zaplanował generelny remont wszystkich peronów, więc część z nich musi zostać wyłączona na pewien czas z użytkowania. Jaka zatem jest minimalna liczba czynnych peronów, która zagwarantuje bezproblemową obsługę wszystkich pociągów (aby żaden pociąg nie czekał)? Opracuj algorytm, który odpowie na to pytanie. Opisz ideę algorytmu a potem zapisz go w pseudokodzie (wraz z niezbędnymi komentarzami). Przeanalizuj czas działania opisanego algorytmu i koniecznie uzasadnij poprawność jego działania.

Zakładamy, że wszystkie pociągi przyjeżdżają i odjeżdżają tego samego dnia; ponadto czas przyjazdu i odjazdu pociągu nigdy nie może być taki sam (każdy pociąg odjeżdża później niż przyjechał). Trzeba także uwzględnić ograniczenie, że jeśli jeden pociąg odjeżdża o tej samej godzinie co przyjeżdża drugi, to nie mogą one używać tego samego peronu (taka sytuacja mogłaby doprowadzić do katastrofy w ruchu lądowym).

Zadanie 2: podział B–drzewa (5 punktów)

Opisz szczegółowo strukturę B–drzewa (z uwzględnieniem parametru t, czyli minimalnego stopnia węzła wewnętrznego).

Następnie opracuj efektywny algorytm rozdzielający B–drzewo T względem zadanej wartości $x \notin T$ na dwa B–drzewa T_l i T_g takie, że wszystkie elementy $< x$ trafią do T_l a wszystkie elementy $> x$ trafią do T_g. Przedstaw ideę algorytmu a potem zapisz go w pseudokodzie (wraz z niezbędnymi komentarzami). Uzasadnij, że opisany algorytm działa poprawnie i przeanalizuj jego złożoność obliczeniową.

Języki formalne i złożoność obliczeniowa

Cofający się automat ze stosem to automat ze stosem mający dodatkowe przejścia postaci $(q, a, b, -1)$, oznaczające, że jeśli automat w stanie q czyta literę a i ma na stosie symbol b, to cofa się o jedną pozycję w słowie wejściowym nie zmieniając stanu ani zawartości stosu. Przyjmujemy, że cofający się automaty ze stosem $\mathcal{A}$ akceptuje słowo w, jeśli osiągnie stan akceptujący będąc na dowolnej pozycji w słowie.

Jaka jest złożoność problemu: dany cofający się automaty ze stosem $\mathcal{A}$, czy istnieje słowo w akceptowane przez $\mathcal{A}$?