Egzamin licencjacki/inżynierski

3 lipca 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

(a) Pokaż, że dla dowolnych zbiorów A, B takich, że $A \cap B = \emptyset$ zbiory $\mathcal{P}(A) \times \mathcal{P}(B)$ oraz $\mathcal{P}(A \cup B)$ są równoliczne.

(b) Pokaż, że dla dowolnych skończonych zbiorów A, B zbiory $\mathcal{P}(A) \times \mathcal{P}(B)$ oraz $\mathcal{P}(A \cup B)$ są równoliczne wtedy i tylko wtedy, gdy $A \cap B = \emptyset$.

(c) Czy stwierdzenie z punktu (b) pozostaje prawdziwe, jeśli słowo „skończonych" zamienimy na „nieskończonych"?

Matematyka II — Algebra

Zadanie 1. (5 punktów)
Wykazać, że wektory własne v_1, v_2 odpowiadające różnym wartościom własnym λ_1, λ_2 są liniowo niezależne.

Zadanie 2. (4 punkty)
Niech $A, B \in \mathbb{R}^{n \times n}$. Udowodnić, że $(AB)^T = B^T A^T$.

Zadanie 3. (5 punktów)
W grupie G elementy a, b są rzędu 2. Wykazać, że $ab = ba$.

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

Metody Programowania

Poniższe zadania należy rozwiązać używając języka Racket lub Plait.

Zadanie 1 Zaproponuj reprezentację formuł rachunku zdań o następującej gramatyce:

$$\psi ::= p \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2$$

gdzie p reprezentuje zmienne zdaniowe. Jeżeli Twoim metajęzykiem jest Racket, to napisz predykat sprawdzający czy jego argument jest poprawną formułą.

Zadanie 2 Napisz procedurę `eval`, która dla danego wartościowania zmiennych zdaniowych (od Ciebie zależy reprezentacja wartościowania), zwraca wartość logiczną formuł z zadania 1.

Zadanie 3 Napisz procedurę `to-nnf`, która przekształca formułę z zadania 1 do negacyjnej postaci normalnej:

$$\begin{aligned} \phi &::= l \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \\ l &::= p \mid \neg p \end{aligned}$$

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** Zadanie z cyklu *Z życia wzięte...*

   - Miejsce: Parter budynku Instytutu Informatyki UWr.
   - Czas: Tuż przed egzaminem z analizy numerycznej (L).
   - Osoby: Balbinka i Ptyś.
   - Opis wydarzenia:
     - Balbinko, już mam! Ten problem z dodatkowej listy zadań, o którym wczoraj mówiłaś można rozwiązać w prosty sposób.
     - No jasne, PWO zawsze daje proste zadania...
     - Ale naprawdę – mówi Ptyś – tym razem w zadaniu nie ma żadnej pułapki. Wystarczy odpowiednio użyć uogólnionego schematu Hornera! Zobacz.
     - Hmm, no niby tak... Zaraz, zaraz – zastanawia się Balbinka – ale skąd wiesz, że wynik będzie bliski temu rzeczywistemu? Przecież nie liczymy dokładnie, a z użyciem komputera i arytmetyki fl. Pewnie podczas obliczeń pojawi się wiele zaokrągleń.
     - No jak to skąd – prawie krzyczy Ptyś – przecież uogólniony schemat Hornera jest algorytmem numerycznie poprawnym! Było o tym na wykładzie, nie pamiętasz? Ta własność wystarczy, żeby mieć pewność, że wynik obliczony przy pomocy komputera i arytmetyki fl będzie OK.

     Balbinka zamyśliła się. Nie trwało to jednak długo. Uśmiechnęła się, złapała Ptysia za rękę i razem z nim radośnie pobiegła w stronę wejścia do sali nr 25. Egzamin zaraz miał się rozpocząć.

   Czy Ptyś miał rację? Czy Balbinka dobrze zrobiła dając się przekonać do zaproponowanego przez kolegę rozwiązania? Odpowiedź **dokładnie** uzasadnij.

2. **4 punkty** Opisz **szczegółowo** metodę bisekcji i jej własności.

3. **4 punkty** Załóżmy, że istnieje rozkład LU macierzy $A := [a_{ij}] \in \mathbb{R}^{n \times n}$. **Wyprowadź wzory**[1] (nie chodzi tu więc o ich napisanie *z pamięci*! – **za to dostaniesz 0 punktów**) na elementy macierzy trójkątnej dolnej $L := [l_{ij}] \in \mathbb{R}^{n \times n}$ oraz macierzy trójkątnej górnej $U := [u_{ij}] \in \mathbb{R}^{n \times n}$, dla których zachodzi $A = L \cdot U$. Następnie **zaproponuj** `algorytm` znajdowania rozkładu LU i **podaj** jego złożoność obliczeniową i pamięciową.

[1] Nie jest to trudne – spróbuj!

Matematyka dyskretna

Niech A będzie podzbiorem zbioru $\{1, 2, \ldots, 150\}$ liczącym 25 elementów. Pokaż, że w A istnieją dwie rozłączne pary elementów o jednakowych sumach.

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: liczba inwersji w permutacji (4 punkty)

Dana jest permutacja, czyli ciąg n różnych liczb ze zbioru $\{1, 2, \ldots, n\}$. Opracuj algorytm, który obliczy liczbę inwersji występujących w takiej permutacji (liczbę różnych par elementów w ciągu, które pozostają w inwersji). Opisz ideę algorytmu a potem zapisz go w pseudokodzie (wraz z niezbędnymi komentarzami). Przeanalizuj złożoność czasową i pamięciową opisanego algorytmu.

Przypomnienie: inwersja w n-elementowym ciągu $(a_1, a_2, \ldots, a_n)$ zachodzi wtedy, gdy dla pary różnych elementów a_i oraz a_j zachodzi nierówność $a_i > a_j$ dla $1 \leq i < j \leq n$ (wcześniejszy element w ciągu jest większy od późniejszego).

Zadanie 2: operacja *next* w drzewie BST (5 punktów)

Dane jest drzewo BST, w którym każdy węzeł przechowuje jedynie pewną wartość liczbową oraz dwa wskaźniki na lewego i prawego syna. Drzewo to jest niemodyfikowalne: można się po nim poruszać (dozwolona jest operacja *search*) ale nie można zmieniać jego zawartości (niedozwolone są operacje *insert* i *delete*) ani struktury (nie można przykładowo realizować operacji *splay* ani nawet prostych rotacji *rotate–left* czy *rotate–right*).

Zadanie polega na zaprojektowaniu operacji *next(x)* na takim drzewie BST, która będzie wyznaczać następną co do wielkości wartość w drzewie — będzie to więc najmniejszą wartość spośród większych od x. Można to zadanie uprościć zakładając, że x zawsze znajduję się w drzewie BST.

Zaprojektuj algorytm, który we wskazanym n-elementowym drzewie BST o wysokości h znajdzie następnik y dla zadanej wartości x. Opisz ideę algorytmu a potem zapisz go w pseudokodzie (wraz z niezbędnymi komentarzami). Przeanalizuj złożoność obliczeniową opisanego algorytmu w zależności od parametrów n (liczba węzłów w drzewie) albo h (wysokość drzewa) albo najlepiej w zależności od głębokości węzłów w drzewie zawierających wartości x i y.

Uwaga: co się stanie, gdy x będzie maksymalną wartością przechowywaną w drzewie BST?

Języki formalne i złożoność obliczeniowa

Rozważmy problem BRI (Bounded Regular Intersection) zdefiniowany następująco:

Dane: dodatnie n, alfabet Σ oraz deterministyczne automaty skończone $A_1, \ldots, A_n$

Pytanie: czy istnieje słowo o długości co najwyżej n, akceptowane przez wszystkie automaty $A_1, \ldots, A_n$ (t.j., należące do $L(A_1) \cap \ldots \cap L(A_n)$)?

Jaka jest złożoność problemu BRI?