Egzamin licencjacki/inżynierski — 6 lutego 2013

Z zestawu sześciu zadań (Matematyka I, Matematyka II, Programowanie, Matematyka dyskretna, Algorytmy i struktury danych i Metody numeryczne) poniżej należy wybrać i przedstawić na osobnych kartkach rozwiązania trzech zadań. Za brakujące (do trzech) zadania zostanie wystawiona ocena nieostateczna z urzędu. Egzamin uważa się za zaliczony, jeśli student rozwiąże z oceną dostateczną co najmniej 2 zadania. Wtedy ocena z egzaminu jest średnią arytmetyczną ocen z trzech wybranych zadań. Na rozwiązanie zadań przeznacza się czas 3x40=120 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

Rozważmy porządek leksykograficzny na parach liczb naturalnych $\langle \mathbb{N}\times\mathbb{N}, \leq_{lex}\rangle$ zdefiniowany formułą

$$\langle x_1, x_2\rangle \leq_{lex} \langle y_1, y_2\rangle \overset{\text{df}}{\Leftrightarrow} x_1 < y_1 \vee (x_1 = y_1 \wedge x_2 \leq y_2).$$

Udowodnij, że $\langle \mathbb{N}\times\mathbb{N}, \leq_{lex}\rangle$ jest dobrym porządkiem. Dla przypomnienia: $\langle X, \leq\rangle$ jest dobrym porządkiem jeśli $\leq$ jest liniowym porządkiem na zbiorze X i każdy niepusty podzbiór X ma element najmniejszy w porządku $\leq$. Nie trzeba dowodzić, że ralecja $\leq_{lex}$ jest relacją porządku.

Matematyka II — Algebra

Zadanie 1.

Sprawdzić, czy wektory $v_1 = \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix}$, $v_2 = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$, $v_3 = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$ są liniowo niezależne
a. w przestrzeni R^3 nad ciałem R,
b. w przestrzeni Z_3^3 nad ciałem Z_2.

Zadanie 2.
Rozważamy grupę $\Phi(12)$ liczb względnie pierwszych z 12, wraz z mnożeniem modulo 12.
i. Sprawdzić, czy jest to grupa cykliczna.
ii. Wyznaczyć rzędy elementów tej grupy.

Zadanie 3.
Znaleźć α_n, β_n takie, że $D_n = \alpha_n D_{n-1} + \beta_n D_{n-2}$.

$$D_n = \begin{vmatrix} a_1 & b_1 & & & \\ c_1 & a_2 & b_2 & & \\ & c_2 & a_3 & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ & & & c_{n-1} & a_n \end{vmatrix}$$

Programowanie

Za zadanie można otrzymać 20 punktów. Aby otrzymać ocenę dostateczną, należy zdobyć 7 punktów, próg dla dst+ to 9p, dla db – 11p, dla db+ 13p, dla bdb – 15p.

Część 1. Gramatyka G_1 z symbolem startowym S nad alfabetem $\{a, b, c\}$ dana jest za pomocą następującego zbioru produkcji:

$$\{C \to \varepsilon,\ C \to cC,\ X \to XX,\ X \to aXb,\ X \to bXa,\ X \to \varepsilon,\ S \to XC\}$$

Gramatyka G_2 z symbolem startowym S nad alfabetem $\{a, b, c\}$ dana jest za pomocą następującego zbioru produkcji:

$$\{A \to \varepsilon,\ A \to aA,\ X \to XX,\ X \to cXb,\ X \to bXc,\ X \to \varepsilon,\ S \to AX\}$$

Dla gramatyki G przez $L(G)$ rozumieć będziemy język generowany przez G. Dla wyrażenia regularnego r przez $\mathcal{L}(r)$ rozumiemy język opisany przez wyrażenie r.

a) Czy *abbbaacc* należy do $L(G_1)$? Odpowiedź uzasadnij. **(1)**

b) Przedstaw wyrażenie regularne lub gramatykę bezkontekstową generującą zbiór $\mathcal{L}(b^*a^*) \cap L(G_1)$ **(2)**

c) Opisz możliwie dokładnie, jakie słowa należą do zbioru $A_1 = L(G_1) \cap L(G_2)$. Odpowiedź krótko uzasadnij. **(3)**

d) Napisz w języku imperatywnym funkcję, która bierze jako wejście napis i zwraca wartość logiczną, równą True wtedy i tylko wtedy, gdy ten napis należy do zbioru A_1. Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, PHP, AWK. **(4)**

Część 2. W tym zadaniu powinieneś się skoncentrować na czytelności kodu, efektywność nie ma żadnego znaczenia. Będziemy używali pojęcia dzielnik jednorodny. *Dzielnikiem jednorodnym* liczby jest każdy jej dzielnik, który jest równy a^b (dla liczby pierwszej a i naturalnej b). Dla liczby 120 dzielnikami jednorodnymi są na przykład 4 albo 5, ale nie 20.

Wariant funkcjonalny. Haskell.

a) Napisz funkcję, która bierze na wejściu dodatnią liczbę całkowitą i zwraca listę jej pierwszych dzielników. Przykładowo, funkcja ta dla liczby 120 powinna zwrócić listę `[2,2,2,3,5]` (lub jakąś jej permutację, bowiem kolejność dzielników nie jest dla nas istotna). **(5)**

b) Jak wykorzystać tę funkcję do napisania funkcji, która sprawdza, czy liczba jest pierwsza? **(1)**

c) Napisz funkcję, która dla danej liczby zwraca jej największy dzielnik jednorodny. **(4)**

Wariant logiczny. Prolog.

a) Napisz predykat `dzielniki(N,L)`, który unifikuje `L` z listą pierwszych dzielników dodatniej całkowitej liczby `N`. Przykładowo, predykat ten dla liczby 120 powinna zunifikować `L` z `[2,2,2,3,5]` (lub jakąś jej permutacją, bowiem kolejność dzielników nie jest dla nas istotna). **(5)**

b) Jak wykorzystać ten predykat, do napisania predykatu, który sprawdza, czy liczba jest pierwsza? **(1)**

c) Napisz predykat `maxJednorodny(N,K)`, która dla danej dodatniej całkowitej liczby `N` unifikuje `K` z jej największym dzielnikiem jednorodnym.

Matematyka dyskretna

Ile wynosi liczba podziałów odcinka o długości n na odcinki o długości 1 i 2?

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: maksymalny wzrost (4 punkty)

Dana jest n–elementowa tablica liczb rzeczywistych $T[0 \ldots n-1]$. Należy wyznaczyć parę indeksów $(i,\, j)$ taką, że $0 \leq i < j < n$ oraz $T[j] - T[i]$ jest maksymalne.

1. Opracuj metodę, która efektywnie rozwiązuje to zadanie; zapisz swój algorytm w pseudokodzie i opisz go krótko.

2. Uzasadnij poprawność opisanego algorytmu; oszacuj jego złożoność obliczeniową (czasową i pamięciową).

Uwaga. Algorytm działający w czasie kwadratowym nie jest efektywnym rozwiązaniem.

Zadanie 2: spójne składowe w grafie (5 punktów)

Dany jest graf prosty $G(V, E)$, gdzie $V = \{0, 1, \ldots, n-1\}$ to zbiór wierzchołków a $E \subseteq \{(i, j) \;:\; i, j \in V \wedge i \neq j\}$ to zbiór krawędzi. Graf ten należy podzielić na *spójne składowe*, nadając każdemu wierzchołkowi dodatkowy numer, mówiący do której spójnej składowej należy.

1. Opracuj algorytm, który efektywnie rozwiązuje to zadanie — w swoim rozwiązaniu wykorzystaj zbiory rozłączne w postaci drzewiastej.

2. Uzasadnij poprawność opisanego algorytmu; oszacuj jego złożoność obliczeniową.

3. Krótko ale precyzyjnie opisz zastosowaną w algorytmie strukturę danych reprezentującą zbiory rozłączne (napisz procedury *union* i *find* w pseudokodzie i oszacuj ich złożoność obliczeniową).

4. Jaka struktura danych najlepiej nadaje się w tym przypadku do pamiętania grafu? Odpowiedź uzasadnij.

Metody numeryczne

1. Niech dana będzie funkcja ciągła f mająca w przedziale (a, b) dokładnie jedno miejsce zerowe α i spełniająca warunek $f(a)f(b) < 0$. Sformułuj i krótko uzasadnij *algorytm bisekcji* znajdowania miejsca zerowego α funkcji f. Ile kroków metody bisekcji należy wykonać, aby wyznaczyć przybliżoną wartość α z błędem bezwzględnym mniejszym niż zadane $\varepsilon > 0$?