Egzamin licencjacki/inżynierski — 26 czerwca 2013

Z sześciu poniższych zestawów zadań (Matematyka I, Matematyka II, Programowanie, Matematyka dyskretna, Algorytmy i struktury danych i 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 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

Niech $M_{2,2013}$ będzie zbiorem macierzy o rozmiarze 2×2013 o współczynnikach rzeczywistych. Rozważmy funkcję $f : M_{2,2013} \to M_{2,2013}$ zdefiniowaną wzorem

$$f(x) \quad = \quad \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} \cdot x$$

gdzie $\cdot$ jest zwykłą operacją mnożenia macierzy. Udowodnij, że funkcja f jest bijekcją.

Matematyka II — Algebra

Za zadania można otrzymać 14 punktów. Aby otrzymać ocenę dostateczną, należy zdobyć 5 punktów, próg dla dst+ to 6.5p, dla db – 8p, dla db+ 9.5p, dla bdb – 11p.

Zadanie 1. (4 punkty)
Dane jest przekształcenie liniowe $T : \mathbb{R}^3 \to \mathbb{R}^3$, określone wzorem

$$T(x_1, x_2, x_3) = (x_1 + x_2,\ 2x_1 - x_3,\ x_1 + x_2 - x_3).$$

Znaleźć macierz tego przekształcenia w bazie $f_1 = (0, 1, 1)$, $f_2 = (1, 0, 1)$, $f_3 = (1, 1, 0)$

Zadanie 2. (6 punktów)
i. Wyznaczyć całkowite liczby a, b takie, że $19a + 13b = 1$.
ii. Rozwiązać równanie $13x \equiv_{19} 1$.
Odpowiedzi uzasadnić.

Zadanie 3. (4 punkty)
Stosując algorytm eliminacji Gaussa wyznaczyć macierze L, U takie, że

$$L\,U = \begin{bmatrix} 1 & 3 & 2 \\ 1 & -1 & 4 \\ 1 & 5 & 0 \end{bmatrix}.$$

L oznacza macierz trójkątną dolną z jedynkami na przekątnej; U to macierz trójkątna górna.

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:

$$\{S \to \varepsilon,\ S \to aSb,\ S \to aSc,\ S \to SS\}$$

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 $aaabbcab$ należy do $L(G_1)$? Odpowiedź uzasadnij. **(1)**

b) Co to znaczy, że gramatyka jest jednoznaczna. Czy gramatyka G_1 jest jednoznaczna (odpowiedź uzasadnij)? **(2)**

c) Dla następujących zbiorów przedstaw wyrażenia regularne je reprezentujące, ewentualnie gramatyki bezkontekstowe je generujące. Przy czym, jeżeli to tylko jest możliwe, powinieneś używać wyrażeń regularnych. **(4)**

- $A_1 = L(G_1) \cap \mathcal{L}(a^*(bc)^*)$.
- $L(G_1) \cap \mathcal{L}(c^*b^*a^*)$
- $L(G_1) \cap \mathcal{L}(a^*b^*c^*)$

d) Napisz w języku wybranym ze zbioru C, C++, Java, C#, Pascal, Python, Ruby, funkcję, która przyjmuje jako argument napis i zwraca wartość logiczną, mówiącą o tym, czy napis należy do A_1. **(3)**

Część 2. W zadaniu będziemy rozważać kompresję listy elementów, polegającą na zamianie ciągu jednakowych wartości na parę, zawierającą tę wartość wraz z liczbą wystąpień. Przykładowo lista:

```
[1,1,1,1,1,1,1,0,0,0,145]
```

kompresuje się do

```
[ (1,7) , (0,3), (145,1)]
```

Twoim celem będzie napisanie funkcji (lub predykatów), który kompresują i rozkompresowywują zadaną listę. Zadanie warte jest 10p. W obu wariantach efektywność nie ma żadnego znaczenia, a istotna jest czytelność. Możesz definiować funkcje (predykaty) pomocnicze, definiując taką funkcję (predykat) powinieneś wyraźnie opisać jej działanie (m.in. podać typ w Haskellu i sposób użycia w Prologu). Jeżeli nie jest to wyraźnie zabronione, możesz korzystać z funkcji (predykatów) standardowych, ale ich działanie również powinieneś opisywać (tak jak tych, które definiujesz samemu).

Wariant funkcjonalny. Haskell.

W punkcie a) nie powinieneś korzystać z żadnych funkcji standardowych

a) Określ typ i zdefiniuj funkcję kompresującą listy. **(5)**

b) Napisz funkcję która bierze jako argument parę składającą się z elementu i liczby N, a zwraca listę zawierającą N krotne powtórzenie tego elementu. **(2)**

c) Określ typ i zdefiniuj funkcję rozkompresowującą uprzednio skompresowane listy. **(3)**

Wariant logiczny. Prolog

W punkcie a) nie powinieneś korzystać z żadnych predykatów standardowych.

a) Określ potrzebne argumenty i tryb użycia, a następnie zdefiniuj predykat kompresujący listy. **(5)**

b) Napisz dwuargumentowy predykat, którego pierwszym argumentem jest para składającą się z elementu i liczby N, a w wyniku działania drugi argument unifikuje się z listą zawierającą N krotne powtórzenie tego elementu. **(2)**

c) Określ potrzebne argumenty i tryb użycia, a następnie zdefiniuj predykat rozkompresowujący uprzednio skompresowane listy. **(3)**

Matematyka dyskretna

Napisz zwarty wzór na sumę

$$\sum_{k=1}^{n}(2k-1)^2.$$

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: kopce dwumianowe (5 punktów)

Opisz strukturę danych zwaną *kopcem dwumianowym*:

- Opisz budowę kopca dwumianowego (w tym drzew dwumianowych).
- Wykaż indukcyjnie, że w drzewie dwumianowym stopnia k długość ścieżki od korzenia do dowolnego węzła jest niewiększa niż k.
- Jakie operacje są efektywnie zaimplementowane w tej strukturze danych i jaka jest ich złożoność czasowa? Opisz dokładniej operację łączenia kopców dwumianowych.
- Narysuj kopiec dwumianowy zawierający 25 różnych liczb naturalnych.
- Opisz algorytm obliczania minimalnego drzewa rozpinającego z wykorzystaniem kopców dwumianowych i oszacuj jego złożoność czasową.

Zadanie 2: suma sześcianów (4 punkty)

Zadana jest liczba naturalna $n > 1$. Liczbę tą należy przedstawić w postaci *sumy sześcianów* liczb naturalnych. Suma ta ma się składać z minimalnej liczby składników.

- Opracuj efektywny algorytm rozwiązujący ten problem.
- Uzasadnij poprawność opisanego algorytmu; oszacuj jego złożoność obliczeniową.
- Wykaż, że strategia zachłanna nie zawsze daje optymalny wynik.

Metody numeryczne

1. Niech dane będą macierze $A, B \in \mathbb{R}^{n\times n}$. Załóżmy, że $\det A \neq 0$. Opracuj oszczędny algorytm wyznaczania takiej macierzy $X \in \mathbb{R}^{n\times n}$, aby zachodziła równość

$$AX = B.$$

   Podaj jego złożoność czasową i pamięciową.