Egzamin licencjacki/inżynierski — 05 lutego 2018

<u>**Informacja dla zdających egzamin na kierunku informatyka:**</u>
Z sześciu poniższych zestawów zadań (Matematyka I, Programowanie, Matematyka dyskretna, Algorytmy i struktury danych, Metody numeryczne, Matematyka II) należy wybrać i przedstawić na osobnych kartkach rozwiązania trzech zestawów.

<u>**Informacja dla zdających egzamin na kierunku indywidualne studia informatyczno-matematyczne:**</u>
Z sześciu poniższych zestawów zadań (Matematyka I, Programowanie, Matematyka dyskretna, Algorytmy i struktury danych, Metody numeryczne, Języki formalne i złożoność obliczeniowa) należy wybrać i przedstawić na osobnych kartkach rozwiązania trzech zestawów.

<u>**Informacja dla wszystkich zdających:**</u>
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 $\langle P, \leq \rangle$ będzie zbiorem uporządkowanym. Powiemy, że funkcja $f : P \to P$ jest w tym porządku

- *rozdmuchująca*, jeśli dla wszystkich $x \in P$ spełniony jest warunek $x \leq f(x)$,
- *monotoniczna*, jeśli dla wszystkich $x, y \in P$ spełniony jest warunek $x \leq y \Rightarrow f(x) \leq f(y)$.

Rozważmy zbiór $\mathbb{N} \times \mathbb{N}$ uporządkowany leksykograficznym rozszerzeniem $\leq_{lex}$ zwykłej relacji porządku na liczbach naturalnych. Udowodnij, że każda różnowartościowa i monotoniczna funkcja $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ jest w tym porządku rozdmuchująca.

Uwaga: To jest zadanie z logiki. Przy ocenianiu zwrócimy szczególną uwagę na poprawność i klarowność rozumowania, odpowiednie sformułowanie i użycie wszystkich założeń, odpowiednie użycie kwantyfikatorów i nawiasów itp.

Programowanie

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

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

$$S \to aSa,\ S \to bSb,\ S \to \varepsilon,\ S \to ccScc,\ S \to aaSaa$$

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

$$S \to bSb,\ S \to cSc,\ S \to \varepsilon,\ S \to aaSa, S \to aSaa$$

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

b) Czy gramatyka G_2 jest jednoznaczna? Odpowiedź krótko uzasadnij. **(2)**

c) Przedstaw możliwie proste wyrażenie regularne lub gramatykę bezkontekstową generującą zbiór $A_1 = L(G_1) \cap L(G_2)$ (ewentualnie napisz, że taka gramatyka nie istnieje). Odpowiedź uzasadnij **(4)**

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 $L(G_1)$. Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, PHP, AWK, Pascal. **(3)**

Zadanie 2. Rozważmy poniższy program w Prologu:

```
p([])
p(L) :-
  append(R, [_,_], L),
  append(A,B,R),
  p(A), p(B).
```

a) Wyjaśnij, co sprawdza ten predykat, gdy go uruchamiamy na liście o ustalonej długości. **(2p)**

b) Predykat ten jest bardzo nieefektywny. Podaj dwie przyczyny tej nieefektywności. **(2p)**

c) Podaj przykład listy o ustalonej długości, dla której zapytanie `?- p(L),!.` wykona się w czasie wielomianowym w zależności od długości L i takie, w którym wykona się w czasie wykładniczym (lub większym). Uzasadnij krótko dlaczego.**(1p)**

d) Napisz efektywny predykat `p2`, który daje te same wyniki co `p1` uruchamiany na liście o ukonkretnionej długości. **(2p)**

Zadanie 3. Napisz w Haskellu funkcję, która bierze listę liczb i zwraca sumę trzycyfrowych liczb parzystych występujących na tej liście. **(3p)**

Matematyka dyskretna

W kwadracie o boku 1 znajduje się 5 punktów. Pokaż, że jakieś dwa z nich są w odległości co najwyżej $1/\sqrt{2}$. Pokaż, też, że gdyby w tezie zastąpić $1/\sqrt{2}$ mniejszą liczbą, to teza nie byłaby prawdziwa.

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: transport pasażerów zepsutego pociągu (4 punkty)

Pewien pociąg utknął w połowie trasy w szczerym polu. Przewoźnik postanowił wynająć taksówki, aby przewieźć zmarzniętych pasażerów do docelowych miejscowości. Pasażerowie pociągu nie chcieliby być rozdzieleni z osobami, z którymi jechali w jednym przedziale. Na szczęście przedziały są czteroosobowe, tak samo jak taksówki.

Zaprojektuj algorytm, który wyliczy, ile najmniej taksówek należy wysłać po pasażerów zepsutego składu, aby byli oni zadowoleni (czyli aby pasażerowie w przedziale nie zostali rozdzieleni do różnych taksówek). Opisz ideę rozwiązania (napisz jaką techniką będziesz rozwiązywał ten problem) a potem dokładnie go opisz (może być pseudokod z wyczerpującymi komentarzami). Uzasadnij poprawność przedstawionej metody i oszacuj jej złożoność obliczeniową.

Zadanie 2: słownik z operacją `between` (5 punktów)

Niech S będzie słownikiem z dodatkową operacją: `between(x, y)`. Chcielibyśmy efektywnie wykonywać następujące operacje na tej strukturze danych:

1. *S = new()* : $S \leftarrow \emptyset$ (utworzenie pustego zbioru);
2. *S.insert(x)* : $S \cup \{x\}$ (dodanie nowego elementu do zbioru);
3. *S.delete(x)* : $S \setminus \{x\}$ (usunięcie elementu ze zbioru);
4. *S.search(x)* : $x \in S$ (sprawdzenie czy element należy do zbioru);
5. *k = S.between(x, y)* : $\#\{z \in S \ : \ x \leq z \leq y\}$ (wyznaczenie liczby elementów w S, których wartości mieszczą się w zakresie od x do y włącznie).

Zaprojektuj taką strukturę danych dla S, która umożliwi wykonywanie wymienionych operacji w czasie logarytmicznym $O(\log n)$, gdzie $n = |S|$ (liczba elementów w S). Możesz wykorzystać jakąś znaną strukturę danych, która efektywnie realizuje operacje słownikowe i zaadoptować ją na potrzeby zadania. Napisz procedurę *between(x, y)* w pseudokodzie i wyjaśnij jak ona działa. Krótko ale precyzyjnie opisz, jak należy zmodyfikować standardowe operacje słownikowe. Przeanalizuj złożoność czasową wszystkich operacji na tej strukturze.

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+ 9 pkt., a dla bdb – 11 pkt.

1. $\boxed{\textbf{4 punkty}}$ Zaproponuj efektywny algorytm obliczania z dużą dokładnością wartości $\sqrt{a}$ $(a > 0)$ wykorzystując **jedynie** operacje arytmetyczne.

2. $\boxed{\textbf{4 punkty}}$ Niech dane będą parami różne liczby rzeczywiste $x_0, x_1, \ldots, x_n$ oraz liczby rzeczywiste $y_0, y_1, \ldots, y_n$. Opisz szczegółowo algorytm **szybkiego** wyznaczania wartości

$$L_n(z_0), L_n(z_1), \ldots, L_n(z_M),$$

gdzie z_i $(0 \le i \le M)$ są danymi liczbami, a M jest **dużą** liczbą naturalną (tj. $M \gg n$), natomiast $L_n \in \Pi_n$ spełnia warunki $L_n(x_i) = y_i$ $(i = 0, 1, \ldots, n)$. Podaj złożoność czasową i pamięciową opracowanej metody.

3. $\boxed{\textbf{4 punkty}}$ Podaj ideę kwadratur złożonych. Wyprowadź wzór na złożony wzór Simpsona.

Matematyka II — Algebra

Zadanie 1. (8 punktów)
Niech G będzie grupą. $Z(G)$ oznacza centrum grupy, to znaczy

$$Z(G) = \{z \in G \mid \forall g \in G \; zg = gz\}.$$

Udowodnić, że centrum grupy jest podgrupą grupy G. Rozpocząć od podania warunków na to aby podzbiór grupy był grupą.

Zadanie 2. (5 punktów)
Sprawdzić, czy wektory $w_1 = [4\ 3\ 2]$, $w_2 = [1\ 1\ 4]$, $w_3 = [2\ 0\ 1]$ są niezależne w przestrzeni liniowej $\mathbb{Z}_7^3$ (przestrzeni nad ciałem dodawania i mnożenia *modulo*) 7.

Progi punktowe: 3, 5, 7, 9, 11 punktów.

Języki formalne i złożoność obliczeniowa

Zadanie 1. (3 punkty) Czy istnieją dwa języki, które nie są bezkontekstowe, a których suma jest językiem regularnym?

Zadanie 2. (2 punkty) Uzasadnij, że każdy problem rozwiązywalny w pamięci wielomianowej można rozwiązać w wykładniczym czasie.

Zaliczenie od 3 punktów, ocena to liczba zdobytych punktów.