Egzamin licencjacki/inżynierski — 15 lutego 2018

Informacja dla zdających egzamin na kierunku informatyka:
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.

Informacja dla zdających egzamin na kierunku indywidualne studia informatyczno-matematyczne:
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.

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=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

1. Sformułuj zasadę indukcji w takiej postaci, żeby można było jej użyć w dowodzie w punkcie 2.

2. Niech $\mathbb{N}_+ = \{n \in \mathbb{N} \mid n \geq 1\}$. Rozważmy funkcję $\mathsf{nwd} : \mathbb{N}_+ \times \mathbb{N}_+ \to \mathbb{N}_+$ zdefiniowaną wzorem

$$\mathsf{nwd}(x, y) = \begin{cases} x, & \text{jeśli} \quad x = y \\ \mathsf{nwd}(x - y, y) & \text{jeśli} \quad x > y \\ \mathsf{nwd}(x, y - x) & \text{wpp} \end{cases}$$

   Korzystając z zasady indukcji sformułowanej w punkcie 1 udowodnij indukcyjnie, że dla wszystkich liczb naturalnych $x, y \geq 1$ obliczanie funkcji $\mathsf{nwd}(x, y)$ się nie zapętla.

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

Programowanie

Za tę część egzaminu 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 $\{a, b, c, d\}$ dana jest za pomocą następującego zbioru produkcji:

$$S \to SS,\ S \to abSba,\ S \to X,\ X \to \varepsilon,\ X \to cXd,\ X \to cXc$$

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

b) Czy gramatyka G_1 jest jednoznaczna? Odpowiedź krótko uzasadnij. **(2)**

c) Przedstaw wyrażenie regularne lub gramatykę bezkontekstową generującą zbiór $A_1 = \mathcal{L}((a + cd)^*) \cap L(G_1)$ Odpowiedź 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 $A_2 = L(G_1) \cap \mathcal{L}((a + b)^*)$. Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, PHP, AWK, Pascal. **(4)**

Zadanie 2. Rozważmy poniższy program w Prologu:

```
p(L, L2) :-
   append(A,[X|B],L),
   append(A, B, L2),
   \+ member(X, L2).

p2(L, R) :-
  p(L, New),!,
  p2(New, R).

p2(L,L).
```

Wyjaśnij jednym zdaniem, co robi predykat `p2`. **(2)**
Jaki będzie wynik zapytania **(2)**:

```
?- p2([2,2,2,5,1,9,8,7,a,b,a,b,1], R).
```

Powiedz, dla jakich list L, wywołanie `?- p2(L,R)` powoduje unifikację `R` z listą pustą? **(1)**

Zadanie 3. Napisz w Haskellu funkcję `moda`, która dla listy liczb znajduję tę, która występuje na liście najwięcej razy (remisy rozwiązuje w dowolny sposób). Możesz definiować funkcje pomocnicze oraz korzystać z funkcji standardowych. Dla każdej funkcji (standardowej lub pomocniczej) podaj jej typ i opisz jednym zdaniem, co ona robi. **(5p)**

Matematyka dyskretna

Ile jest w grafie K_n dróg, których jednym z końców jest któryś z wierzchołków v_1 lub v_2 należących do K_n?

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: najdłuższy wspólny podciąg trzech ciągów (4 punkty)

Dane są trzy n-elementowe ciągi znaków $X = (x_1, x_2, \ldots x_n)$, $Y = (y_1, y_2, \ldots y_n)$ i $Z = (z_1, z_2, \ldots z_n)$, gdzie $x_i, y_i, z_i \in \{A, B, \ldots Z\}$ dla $i = 1 \ldots n$. Zaprojektuj algorytm, który wyliczy długość najdłuższego wspólnego podciągu dla ciągów X, Y i Z. Opisz ideę rozwiązania (napisz jaką techniką będziesz rozwiązywał ten problem) a potem dokładnie opisz algorytm (może być pseudokod z wyczerpującymi komentarzami). Uzasadnij poprawność przedstawionej metody i oszacuj jej złożoność obliczeniową.

Zadanie 2: kolejka priorytetowa z operacją `median` (5 punktów)

Zaprojektuj strukturę danych, która będzie efektywnie realizowała następujące operacje na zbiorze dynamicznym: *insert* (wstawienie nowego elementu do zbioru) oraz *extract–median* (usunięcie ze zbioru elementu środkowego co do wielkości, czyli mediany). Opisz dokładnie strukturę danych, która umożliwi wykonywanie wymienionych operacji w czasie logarytmicznym $O(\log n)$, gdzie n jest liczbą elementów przechowywanych w zbiorze. Napisz procedury *insert* i *extract–median* w pseudokodzie i wyjaśnij jak one działają. Przeanalizuj złożoność czasową tych operacji.

Możesz wykorzystać jakąś znaną strukturę danych, która efektywnie realizuje operacje kolejkowe i zaadoptować ją na potrzeby tego zadania.

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. $\boxed{\textbf{3 punkty}}$ Do rozwiązania zadania obliczeniowego $\mathcal{A}$ użyto algorytmu numerycznie poprawnego. Czy można mieć pewność, że otrzymany w ten sposób wynik jest bliski rzeczywistego rozwiązania zadania $\mathcal{A}$? Odpowiedź uzasadnij.

2. $\boxed{\textbf{3 punkty}}$ Pomiary (t_k, c_k) $(0 \le k \le N;\ t_k > 0,\ c_k > 1)$ pewnej zależnej od czasu wielkości fizycznej C sugerują, że wyraża się ona wzorem
$$C(t) = 2^{(At^2+2018)^{-1}}.$$
Stosując aproksymację średniokwadratową, wyznacz prawdopodobną wartość parametru A.

3. $\boxed{\textbf{6 punktów}}$ Niech dane będą wektory $\mathtt{x}{:=}[x_0, x_1, \ldots, x_n]$ ($x_0 < x_1 < \cdots < x_n$), $\mathtt{y}{:=}[y_0, y_1, \ldots, y_n]$ oraz $\mathtt{z}{:=}[z_0, z_1, \ldots, z_m]$. Niech s_n oznacza naturalną funkcję sklejaną trzeciego stopnia (*w skrócie*: NFS3) spełniającą warunki $s_n(x_k) = y_k$ ($0 \le k \le n$). Jak już wiadomo, w języku `PWO++` procedura `NSpline3(x,y,z)` wyznacza wektor $\mathtt{Z} := [s_n(z_0), s_n(z_1), \ldots, s_n(z_m)]$, z tym, że **musi być** $m < 2n$. Załóżmy, że wartości pewnej funkcji ciągłej f znane są **jedynie** w punktach $x_0 < x_1 < \cdots < x_{100}$. Wiadomo, że NFS3 odpowiadająca danym $(x_k, f(x_k))$ ($0 \le k \le 100$) bardzo dobrze przybliża funkcję f. Wywołując procedurę `NSpline3` **tylko raz**, opracuj algorytm numerycznego wyznaczania przybliżonych wartości wszystkich **rozwiązań** równiania $f(x) = g$ znajdujących się w przedziale $[x_0, x_{100}]$, gdzie g jest daną liczbą rzeczywistą. W swoim rozwiązaniu możesz **użyć wielokrotnie** innej procedury języka `PWO++`, a mianowicie `Solve3(a,b,c,d)` znajdującej z dużą dokładnością wszystkie rzeczywiste miejsca zerowe wielomianu $\mathtt{a}x^3 + \mathtt{b}x^2 + \mathtt{c}x + \mathtt{d}$ albo informującej, że takich miejsc zerowych nie ma.

Matematyka II — Algebra

Zadanie 1. (8 punktów)
Niech G będzie grupą. $C(x)$ oznacza centralizator elementu $x \in G$, to znaczy

$$C(x) = \{g \in G \mid gx = xg\}.$$

Podać warunki na to aby podzbiór grupy był jej podgrupą. Udowodnić, że centralizator (dowolnego elementu x) jest podgrupą grupy G.

Zadanie 3. (5 punktów)
Sprawdzić, czy wektory $w_1 = [1\ 2\ 2]$, $w_2 = [2\ 1\ 2]$, $w_3 = [3\ 1\ 1]$ są niezależne w przestrzeni liniowej $\mathbb{Z}_5^3$ (przestrzeni nad ciałem dodawania i mnożenia *modulo* 5.

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