# Egzamin licencjacki/inżynierski

9 lutego 2021

## 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 ISIM

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

Rozważmy krotkę $\langle V, E, c\rangle$ gdzie $V$ jest skończonym zbiorem zwanym zbiorem wierzchołków, $E \subseteq V \times V$ jest binarną relacją na zbiorze $V$ zwaną zbiorem krawędzi, a $c : V \to C$ jest funkcją zwaną kolorowaniem wierzchołków. Powiemy, że

- $\langle V, E\rangle$ jest *grafem prostym* gdy relacja $E$ jest antyzwrotna i symetryczna;
- $\langle V, E\rangle$ jest *pełnym skojarzeniem* gdy jest grafem prostym i dodatkowo relacja $E$ jest funkcją (intuicyjnie: każdy wierzchołek ma dokładnie jednego sąsiada);
- $\langle V, E, c\rangle$ jest *dobrze pokolorowany* gdy spełniony jest warunek

  $$\forall x\ y.\ E(x, y) \Rightarrow c(x) \neq c(y)$$

  (intuicyjnie: każda krawędź ma końce różnych kolorów);
- $\langle V, E, c\rangle$ jest *dobrze pokolorowanym pełnym skojarzeniem* gdy $\langle V, E\rangle$ jest pełnym skojarzeniem i $\langle V, E, c\rangle$ jest dobrze pokolorowany;
- $k \in C$ jest *kolorem dominującym* gdy spełniony jest warunek

  $$|c^{-1}[\{k\}]| > \frac{|V|}{2}$$

  (intuicyjnie: ponad połowa wierzchołków na kolor $k$);
- zbiór $V$ jest *równomiernie pokolorowany* funkcją $c$ gdy spełniony jest warunek

  $$\forall k \in C.\ |c^{-1}[\{k\}]| \leq \frac{|V|}{2}$$

  (intuicyjnie: żaden kolor nie jest dominujący).

Udowodnij indukcyjnie, że dla każdego zbioru wierzchołków $V$ o parzystej mocy, równomiernie pokolorowanego funkcją $c$, istnieje taki zbiór krawędzi $E$, że $\langle V, E, c\rangle$ jest dobrze pokolorowanym pełnym skojarzeniem. Sformułuj zasadę indukcji, z której korzystasz w dowodzie.

*Wskazówka:* szukane skojarzenie można skonstruować algorytmem zachłannym wybierając przedstawicieli najbardziej licznych kolorów.

*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. Nie interesują nas nieindukcyjne rozwiązania tego zadania.

## Programowanie – wariant A

Za tę część egzaminu 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.

**Zadanie 1. (10p)** Poniższe zadanie możesz wykonać w wybranym języku funkcyjnym (Haskell, Ocaml, Racket). Nie używaj przy tym imperatywnych cech języka.

1. Zdefiniuj typ danych reprezentujący drzewa binarne z etykietami w wierzchołkach wewnętrznych. **(2p)**
2. Wyjaśnij pojęcie *nieużytku*. **(2p)**
3. Powiemy, że drzewo binarne jest *zbalansowane*, gdy dla dowolnych dwóch ścieżek od korzenia do liścia ich długość różni się o co najwyżej 1. Zaimplementuj procedurę sprawdzającą czy dane drzewo jest zbalansowane i procedurę, która dla danej listy `xs` generuje zbalansowane drzewo binarne, którego lista etykiet w porządku *preorder* jest równa liście `xs`. Uzasadnij, że żadna z powyższych procedur nie generuje nieużytków. Oczekiwana złożoność obydwu procedur jest liniowa względem rozmiaru danych wejściowych. **(6p)**

**Zadanie 2. (10p)** Poniższe zadanie wykonaj w wybranym języku imperatywnym (np. C, Python, etc.). Nie używaj przy tym funkcji rekurencyjnych ani innych funkcyjnych cech języka.

1. Zdefiniuj typ danych reprezentujący drzewa binarne z etykietami w wierzchołkach wewnętrznych (tak, ten sam). Tym razem możesz ograniczyć się do drzew etykietowanych liczbami. **(2p)**
2. Przypomnijmy, że drzewo binarne nazywamy *drzewem przeszukiwań binarnych* (BST), jeśli jest etykietowane liczbami, a ciąg etykiet wypisany w porządku *inorder* jest posortowany. Zaimplementuj procedurę sprawdzającą czy dane drzewo jest drzewem przeszukiwań binarnych. Oczekiwana złożoność to $O(n)$, gdzie $n$ jest rozmiarem drzewa. **(4p)**
3. Zaimplementuj procedurę usuwania z drzewa przeszukiwań binarnych najmniejszego elementu. Wskaż obiekty, które będzie musiał usunąć system zarządzania pamięcią, jeśli takie pojawią się w Twojej implementacji. **(4p)**

## Programowanie – wariant B

Za tę część egzaminu 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.

**Zadanie 1. (10p)** 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. **(1p)**

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

c) Przedstaw wyrażenie regularne lub gramatykę bezkontekstową generującą zbiór $A_1 = \mathcal{L}((a + cd)^*) \cap L(G_1)$. Odpowiedź uzasadnij. **(3p)**

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 $A_2 = L(G_1) \cap \mathcal{L}((a + b + c)^*)$. Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, AWK, Rust. **(4p)**

**Zadanie 2. (5p)** 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.

**Zadanie 3. (5p)** Napisz w Prologu predykat `my_is`, który działa tak, jakby był zdefiniowany w następujący sposób:

```
my_is(V, E) :- V is E.
```

Predykat ten powinien obsługiwać 4 podstawowe działania. W definicji nie możesz korzystać ze standardowego predykatu `is` (i innych arytmetycznych predykatów prologowych), ale możesz używać następujących, zdefiniowanych niżej predykatów:

```
add(X,Y,Z)   :- number(X), number(Y), Z is X+Y.
mult(X,Y,Z)  :- number(X), number(Y), Z is X*Y.
div(X,Y,Z)   :- number(X), number(Y), Z is X/Y.
minus(X,Y,Z) :- number(X), number(Y), Z is X-Y.
```

## Matematyka dyskretna

Nie korzystając z Tw. Kuratowskiego pokaż, że $K_{3,3}$ nie jest planarny.

## 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 obliczeniowe, o którym wiadomo, że jest dobrze uwarunkowane (przypomnij, co to oznacza) rozwiązujemy przy pomocy algorytmu numerycznie poprawnego (podaj jego definicję). Czy wyznaczony w ten sposób wynik jest wiarygodny? Odpowiedź uzasadnij.

2. $\boxed{\textbf{4 punkty}}$ Niech dana będzie funkcja ciągła $f$ mająca w przedziale $(a, b)$ dokładnie jedno miejsce zerowe $\alpha$ i spełniająca warunek $f(a)f(b) < 0$. Sformułuj i krótko uzasadnij *algorytm bisekcji* znajdowania miejsca zerowego $\alpha$ funkcji $f$. Ile kroków metody bisekcji należy wykonać, aby wyznaczyć przybliżoną wartość $\alpha$ z błędem bezwzględnym mniejszym niż zadane $\varepsilon > 0$?

3. $\boxed{\textbf{4 punkty}}$ Niech dana będzie funkcja ciągła $f$, liczby rzeczywiste $a < b$ oraz liczba naturalna $m$. W języku `PWO++` procedura `RombergTable(`$f, a, b, m$`)` znajduje $m$-ty wiersz tablicy Romberga przybliżeń wartości całki

$$I := \int_a^b f(x)\mathrm{d}x,$$

czyli elementy $T_{0,m-1}, T_{1,m-2}, \ldots, T_{m-1,0}$ tej tablicy. Niestety, procedura ta ma **pewne ograniczenie**, mianowicie: **musi być** $m < 50$. W jaki sposób, używając procedury `RombergTable` **tylko raz** można **efektywnie** wyznaczyć wszystkie elementy 50-tego wiersza tablicy Romberga odpowiadającego całce $I$?

## 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: wydawanie reszty różnymi nominałami (4 punkty)

Rozważmy następującą wersję problemu wydawania reszty. Danych jest $n$ nominałów monet $x_1$, $x_2$, $\ldots$, $x_n$ (nominały są liczbami naturalnymi) oraz pewna kwota $v$. Zadanie polega na uzbieraniu kwoty $v$, wykorzystując każdy nominał co najwyżej raz.

Opracuj algorytm rozwiązujący to zadanie w czasie pseudowielomianowym $O(nv)$. Opisz ideę algorytmu, a potem zapisz go w pseudokodzie (wraz z niezbędnymi komentarzami). Uzasadnij, że opisany algorytm działa poprawnie.

### Zadanie 2: poprzednik elementu w B–drzewie (5 punktów)

Opisz szczegółowo strukturę B–drzewa.

Następnie opracuj efektywny algorytm wyznaczający poprzednika pewnego klucza przechowywanego w B–drzewie. Przedstaw ideę tego algorytmu, a potem zapisz go w pseudokodzie (wraz z niezbędnymi komentarzami). Uzasadnij, że opisany algorytm działa poprawnie i przeanalizuj jego złożoność obliczeniową.

## Matematyka II — Algebra

**Zadanie 1. (4p)**

$$A = \begin{bmatrix} 6 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 4 & 9 \end{bmatrix}.$$

Sprawdzić czy $A$ jest macierzą dodatnio określoną.

**Zadanie 2. (4p)** Obliczyć nwd (największy wspólny dzielnik) liczb 3960 oraz 80850. Przedstawić ten dzielnik jako kombinację liniową (o współczynnikach całkowitych) wspomnianych liczb.

**Zadanie 3. (6p)**

$$J = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda & 1 \\ 0 & 0 & \cdots & 0 & \lambda \end{bmatrix}.$$

$J \in \mathbb{R}^{n \times n}$. Podać krotności algebraiczne i geometryczne wartości własnych tej macierzy.

Progi punktowe: 4 punkty – ocena dst, 6 – dst+, 8 – db, 10 – db+, 12 – bdb.

## Języki formalne i złożoność obliczeniowa

*Słowo nieskończone* nad alfabetem $\Sigma$ to dowolny, nieskończony ciąg elementów $\Sigma$. Deterministyczny automat skończony działa na słowach nieskończonych tak samo, jak na słowach skończonych, tyle, że nigdy się nie zatrzymuje. Powiemy, że automat akceptuje dane słowo nieskończone, jeśli w czasie czytania tego słowa, od pewnego momentu jest już cały czas w stanie akceptującym. Język słów nieskończonych takiego automatu to zbiór tych słów, które automat akceptuje.

1. Czy klasa języków słów nieskończonych rozpoznawanych przez deterministyczne automaty skończone jest zamknięta na przecięcie? **(1p)**
2. Czy język słów nieskończonych nad alfabetem $\Sigma = \{a, b\}$, które mają nieskończenie wiele $b$, daje się rozpoznawać deterministycznym automatem skończonym? **(2p)**
3. Czy następujący problem można rozwiązać w czasie wielomianowym: dla danego deterministycznego automatu skończonego, czy istnieje słowo nieskończone akceptowane przez ten automat? **(2p)**

*Ocena to liczba zdobytych punktów.*