Warszawska Szkoła Doktorska Matematyki i Informatyki

9 czerwca 2022

Egzamin Kwalifikacyjny

Na kolejnych stronach znajduje się 16 zadań dotyczących różnych obszarów matematyki i informatyki. Należy wybrać i rozwiązać dowolne 4 z nich. Każde zadanie jest warte tyle samo punktów.

Zadania można wybierać dowolnie, tj. kandydaci na studia w dyscyplinie matematyka mogą rozwiązywać także zadania "informatyczne" i na odwrót.

Każde zadanie składa się z kilku podzadań o porównywalnej wielkości, jednak każde zadanie (tj. wszystkie jego podpunkty) jest punktowane jako całość.

Możesz spróbować rozwiązać więcej niż 4 zadania. Wszystkie te rozwiązania będą ocenione, jednak tylko 4 najlepiej ocenione zadania wejdą w skład Twojej ogólnej oceny.

Wszystkie odpowiedzi należy odpowiednio uzasadnić.

Każde zadanie należy rozwiązywać na osobnej kartce podpisanej imieniem i nazwiskiem i oznaczonej numerem zadania.

Czas egzaminu: 3 godziny.

Powodzenia!

1. **Algebra liniowa**

   Niech V będzie skończenie wymiarową przestrzenią wektorową nad ciałem liczb rzeczywistych z iloczynem skalarnym $\langle \cdot, \cdot \rangle$ i niech $\mathrm{End}(V)$ będzie zbiorem endomorfizmów V.

   a) Pokazać, że przekształcenie $S : \mathrm{End}(V) \to \mathrm{End}(V)$ określone wzorem $S(T) = T^*$ jest liniowe i spełnia $S(TW) = S(W)\,S(T)$.

   b) Wykazać, że wzór $(T, W) := Tr(T^* W)$ definiuje dodatnio okreslony iloczyn skalarny na $\mathrm{End}(V)$, gdzie Tr oznacza ślad.

   c) Czy przekształcenie S jest samosprzężone?

   d) Opisać wartości i przestrzenie własne przekształcenia S.

   e) Załóżmy, że $A \in \mathrm{End}(V)$ jest samosprzężone, a $T \in \mathrm{End}(V)$ ma ma tę własność, że każdy wektor własny A jest jednocześnie wektorem własnym T. Czy $AT = TA$? Czy założenie samosprzężoności A jest istotne?

   f) Załóżmy, że T spełnia $T^2 = T$. Pokazać, że T jest rzutem ortogonalnym wtedy i tylko wtedy gdy T jest samosprzężone.

2. **Algebra**

   Rozważmy pierścień $R = \mathbb{Z}_{18}[x]$.

   a) Ile jest w pierścieniu R elementów nilpotentnych postaci $a\,x^{10} + bx^2 + c$, gdzie $a, b, c \in \mathbb{Z}_{18}$?

   b) Ile jest w pierścieniu R elementów odwracalnych postaci $sx^7 + u$, gdzie $s, u \in \mathbb{Z}_{18}$? Podać przykład takich $s, u \in \mathbb{Z}_{18}$, $s \neq 0$, że element $sx^7 + u$ jest odwracalny w R oraz podać element do niego odwrotny.

   c) Udowodnić, że ideał $I = (x^2 + 1)$ nie jest pierwszy w R. Wskazać jakikolwiek ideał maksymalny zawierający I.

3. **Analiza**

   Rozważmy następujący układ równań

$$\begin{cases} u \ln x + y e^{u+w} = e, \\ e^{2w} + w \ln(xyz) = e^{-2}. \end{cases} \qquad (\star)$$

   Niech:

$$B_r^3 = \left\{ (x, y, z) \in \mathbb{R}^3 : (x-1)^2 + (y-1)^2 + (z-1)^2 < r^2 \right\}$$

   a) Wykazać, że układ $(\star)$ w otoczeniu punktu $p = (x_0, y_0, z_0, u_0, w_0) = (1, 1, 1, 2, -1)$ wyznacza jednoznacznie zmienne u i w jako funkcje zmiennych (x, y, z). Uzasadnić, że istnieje $R > 0$, takie że funkcje $u(x, y, z)$ oraz $w(x, y, z)$ wyznaczone przez układ $(\star)$ są klasy C^2 na kuli B_R^3.

   b) Wyznaczyć pochodne cząstkowe $\dfrac{\partial u}{\partial y}(1, 1, 1)$ oraz $\dfrac{\partial w}{\partial z}(1, 1, 1)$.

   c) Niech $f \colon \mathbb{R}^2 \to \mathbb{R}$ będzie funkcją różniczkowalną, taką że

$$\mathrm{grad} f(1, -2) = [e, -e].$$

   Niech $h(x, y, z) = f\big(u(x, y, z), w(x, y, z)\big)$.

   - Czy funkcja h ma ekstremum lokalne w punkcie $(1, 1, 1)$?
   - Czy funkcja h przyjmuje swoje kresy na zbiorze $\bar{B}^3_{R/2}$, gdzie $\bar{A}$ oznacza domknięcie zbioru A?

d) Określmy formę różniczkową $\omega \in \Omega^2(B_R^3)$ wzorem

$$\omega(x,y,z) = \frac{\partial u(x,y,z)}{\partial x}\,\mathrm{d}x \wedge \mathrm{d}y - \frac{\partial u(x,y,z)}{\partial z}\,\mathrm{d}y \wedge \mathrm{d}z.$$

Obliczyć dla $r < R$ wartość całki po półsferze

$$\int_{S_r^+} \omega,$$

gdzie $S_r^+ = \{(x,y,z) \in \mathbb{R}^3 : (x-1)^2 + (y-1)^2 + (z-1)^2 = r^2, \quad y \geq 1\}$. Orientacja S_r^+ jest zadana jest przez ciągłe pole wektorów normalnych $\overline{n}$ takie, ze $\overline{n}(1, 1+r, 1) = (0,1,0)$.

4. **Topologia**

Przestrzeń metryczną (X,d) nazywamy ultrametryczną jeśli, dla $x,y,z \in X$, metryka d spełnia następującą (silniejszą) wersję nierówności trójkąta:

$$d(x,z) \leq \max(d(x,y), d(y,z)).$$

a) Pokazać, że dla ustalonej liczby pierwszej p, metryka p-adyczna d na zbiorze liczb całkowitych $\mathbb{Z}$ zdefiniowana przez

$$d(a,b) = \min\{2^{-i} : p^i \text{ dzieli } |a-b|\},$$

dla $a \neq b \in \mathbb{Z}$, jest ultrametryką. Czy w $(\mathbb{Z}, d)$ istnieją punkty skupienia?

b) Pokazać, że jeśli B, B' są kulami w przestrzeni ultrametrycznej o tym samym promieniu $r > 0$, to $B = B'$ lub $B \cap B' = \emptyset$. Wykazać, że zbiór odległości $O = \{d(x,y) : x,y \in X\}$ zwartej przestrzeni ultrametrycznej X jest skończony albo ma postać ciągu zbiegającego do 0.

c) Korzystając z pierwszej części b), scharakteryzować spójne podprzestrzenie przestrzeni ultrametrycznej.

d) Pokazać, że jeśli przeliczalna przestrzeń metryczna X (nie musi być ultrametryczna) jest zupełna to ma punkt izolowany.

5. **Prawdopodobieństwo**

Z urny, zawierającej jedną białą, jedną czarną i jedną zieloną kulę, losujemy kolejno po jednej kuli ze zwracaniem. Losowanie przerywamy w momencie wyciągnięcia czarnej kuli po raz pierwszy.

a) Jakie jest prawdopodobieństwo, że liczba losowań przekroczy 10?

b) Jakie jest prawdopodobieństwo, że kulę białą i kulę zieloną wyciągnięto tyle samo razy?

c) Obliczyć wartość oczekiwaną liczby tych losowań, w których wyciągnięto kulę białą.

d) Obliczyć wartość oczekiwaną łącznej liczby losowań, jeśli wiadomo, że kula zielona nie została wyciągnięta ani razu.

e) Obliczyć prawdopodobieństwo, że losowano 7 razy jeśli wiadomo, że kulę białą wyciągnięto tyle samo razy co kulę zieloną.

6. **Analiza funkcjonalna**

Rozważmy transformatę całkową

$$Tf(x) = \int_{-\infty}^{+\infty} e^{-t^4 x} f(t)\, dt.$$

a) Znaleźć $Tf(x)$ dla $f(t) = t$ oraz $x > 0$.

b) Załóżmy, że $f \in L_p(\mathbb{R})$ dla pewnego $1 \leq p \leq \infty$. Wykazać, że $Tf(x)$ jest dobrze określone dla $x > 0$;

c) Wykazać, że jeżeli $f \in L^\infty(\mathbb{R})$ to

$$|Tf(x)| \leq C x^{-\frac{1}{4}},$$

gdzie stała C zależy tylko od f.

d) Wykazać, że jeżeli $f \in L^p(\mathbb{R})$ dla pewnego $1 < p < \infty$, to

$$|Tf(x)| \leq C x^{-\frac{p-1}{4p}},$$

gdzie stała $C = C(f, p)$.

e) Załóżmy, że $f \in L_p(\mathbb{R})$ dla pewnego $1 \leq p \leq \infty$. Wykazać, że $Tf \in C^1((0, +\infty))$. Znaleźć wzór na $(Tf)'(x)$.

7. **Równania różniczkowe**

a) Wyznaczyć wszystkie rozwiązania zagadnienia Cauchy'ego

$$\begin{cases} x'(t) = x^{3/4} & \text{dla} \quad t > 0, \\ x(0) = 0 \end{cases}$$

b) Znaleźć rozwiązanie ogólne równania

$$x'' - 2x' + x = t, \quad x(0) = 1.$$

Czy istnieje takie rozwiązanie $x(t)$ tego równania, że $x(1) = 1$?

c) Rozważmy zagadnienie

$$\begin{cases} x'(t) = ax^6 - bt & \text{dla} \quad t > 0, \\ x(0) = x_0 > 0, \end{cases}$$

gdzie a i b są dowolnymi, dodatnimi stałymi. Wykaż, że istnieje $x^* > 0$ takie, że jeżeli $x_0 < x^*$ to rozwiązanie jest określone dla wszystkich $t > 0$.

8. **Funkcje analityczne**

a) Wykazać, że $h(z) = \frac{1}{z-1}$ jest jedyną funkcją holomorficzną na $\mathbb{C} \setminus \{1\}$ spełniającą nastepujace warunki

- odwzorowanie $h\colon \mathbb{C} \setminus \{1\} \to \mathbb{C} \setminus \{0\}$ jest wzajemnie jednoznaczne,
- $h(0) = -1$.

b) Niech $n \in \mathbb{N}$, $F(z) = h(z^n)$, $C_R = \{z \in \mathbb{C} : |z| = R\}$, $I_R = \int_{C_R} F(z)dz$ (na okręgu C_R przyjmujemy orientację przeciwną do ruchu wskazówek zegara).
Obliczyć I_R, gdy $R < 1$.

c) Obliczyć I_R dla $R > 1$ i $n = 1$.

d) Obliczyć I_R dla $R > 1$ i $n \geq 2$.

9. **Matematyka dyskretna**

Dany jest graf nieskierowany w postaci kraty, gdzie zbiór wierzchołków to $\{(i,j)\colon i,j \in \{0,1,\ldots,n\}\}$. Mamy $2n(n+1)$ krawędzi łączących sąsiednie punkty: dla każdego $i \in \{0,1,\ldots,n\}$ oraz $j \in \{0,1,\ldots,n-1\}$ mamy krawędź od (i,j) do $(i,j+1)$ oraz od (j,i) do $(j+1,i)$.

(a) Ile jest najkrótszych ścieżek łączących wierzchołki $(0,0)$ i (n,n)?

(b) Ustalmy trzy krawędzie: (1) e_1 łączącą wierzchołki (i_1,j_1) oraz (i_1+1,j_1), (2) e_2 łączącą wierzchołki (i_2,j_2) oraz (i_2+1,j_2) oraz (3) e_3 łączącą wierzchołki (i_3,j_3) oraz (i_3+1,j_3). Mamy $0 \leq i_1 < i_2 < i_3 < n$ oraz $0 \leq j_1 < j_2 < j_3 \leq n$. Ile jest najkrótszych ścieżek łączących wierzchołki $(0,0)$ i (n,n) które przechodzą przez co najmniej jedną z krawędzi e_1, e_2 i e_3? Odpowiedź wystarczy podać w postaci sumy wyrażeń.

(c) Ustalmy podzbiór wierzchołków w postaci kwadratu $S = \{(i,j)\colon i,j \in \{n_\ell, n_\ell+1,\ldots,n_u\}\}$, gdzie $0 < n_\ell < n_u < n$. Ile jest najkrótszych ścieżek łączących wierzchołki $(0,0)$ i (n,n) które przechodzą przez co najmniej jeden punkt z S. Odpowiedź wystarczy podać w postaci sumy wyrażeń.

(d) Ustalmy rosnącą sekwencję $m+1$ liczb naturalnych $k_0,k_1,\ldots,k_m$, gdzie dla każdego $i \in \{0,1,\ldots,m\}$ mamy $k_i \in \{0,1,\ldots,n\}$. Udowodnij następującą tożsamość:

$$\sum_{i=0}^{m} \binom{n-k_i+i}{i}\binom{m+k_i-i}{k_i} \leq \binom{n+m}{n}.$$

Kiedy zachodzi równość?

10. **Złożoność obliczeniowa**

Problem VertexCover jest zdefiniowany następująco. Na wejściu mamy dany nieskierowany graf $G=(V,E)$ oraz liczbę k. Powiemy, że podzbiór wierzchołków $S \subseteq V$ pokrywa wierzchołek $v \in V$, jeżeli $v \in S$ lub jeżeli v jest połączony krawędzią z jakimś wierzchołkiem z S. Pytamy, czy istnieje podzbiór wierzchołków $S \subseteq V$ o rozmiarze k ($|S| = k$) taki, który pokrywa wszystkie wierzchołki grafu G. W wersji optymalizacyjnej, pytamy o podzbiór wierzchołków $S \subseteq V$ o rozmiarze k, który pokrywa jak największą liczbę wierzchołków z G. Problem VertexCover jest NP-trudny.

(a) Udowodnij, że problem VertexCover pozostaje NP-trudny, nawet gdy ograniczymy się do grafów, w których stopień każdego wierzchołka wynosi co najmniej 2022.

(b) Załóżmy, że graf wejściowy ma następującą własność: Jeżeli wypiszemy stopnie wszystkich wierzchołków, to otrzymamy co najmniej $|V| - 2022$ różnych liczb. Czy dla takich grafów problem VertexCover pozostaje NP-trudny?

(c) Rozważmy następujący algorytm lokalnego wyszukiwania. Rozpoczynamy od dowolnego zbioru wierzchołków $S \subseteq V$, $|S| = k$. Następnie w pętli wykonujemy następującą operację. Jeżeli istnieje para wierzchołków $v \in S$ i $v' \notin S$ taka, że zbiór $(S \setminus \{v\}) \cup \{v'\}$ pokrywa więcej wierzchołków niż S, to dokonujemy wymiany (usuwamy v z S i dokładamy v' do S; $S := (S \setminus \{v\}) \cup \{v'\}$). Jeżeli nie istnieje taka para wierzchołków, to przerywamy pętlę i zwracamy zbiór S. Udowodnij, że jest to algorytm 2-aproksymacyjny dla optymalizacyjnej wersji problemu VertexCover.

(d) Rozważmy parametr p który jest zdefiniowany następująco: 2022 wierzchołków ma stopnień większy niż $|V| - p$, a pozostałe mają mniejszy niż p. Czy problem VertexCover jest w klasie FPT (fixed-parameter tractable) dla parametru p? Uzasadnij odpowiedź.

11. **Algorytmy i struktury danych**

Dane jest drzewo z wagami krawędzi $w(e) > 0$ dla $e \in E(T)$. Niech L to zbiór wszystkich liści w T. Dla $X \subseteq L$ niech $T|X$ to najmniejszy spójny podgraf T zawierający wszystkie liście z X. Waga $T|X$ to $w(X) = \sum_{e \in E(T|X)} w(e)$. Dla $k > 0$ niech $p_k(T) = \max\{w(X) \colon X \subseteq L \text{ i } |X| = k\}$.

(a) Pokaż, że dla zbiorów $X, X' \subseteq L$ z $|X| > |X'| \geq 2$, istnieje $x \in X \setminus X'$, takie że,

$$w(X \setminus \{x\}) + w(X' \cup \{x\}) \geq w(X) + w(X'). \tag{1}$$

(b) Wywnioskuj z (1), że jeśli $w(X) = p_k(T)$ dla $2 \leq |X| = k < n$, to istnieje x taki, że $w(X \cup \{x\}) = p_{k+1}(T)$.

(c) Na bazie poprzedniej obserwacji zaproponuj algorytm działający w czasie $O(n^2)$ obliczający $p_k(T)$, gdzie n to rozmiar drzewa T.

(d) Zaproponuj algorytm obliczający $p_k(T)$ działający w czasie $O(n \log n)$.

12. **Logika i bazy danych**

Czy następujące problemy decyzyjne są rozstrzygalne?

(a) Czy dane zdanie logiki pierwszego rzędu (nad dowolną sygnaturą) ma model 2022-elementowy?

(b) Czy dane zdanie logiki pierwszego rzędu nad sygnaturą złożoną z trójargumentowego symbolu relacyjnego *triangle* (i równości) jest prawdziwe w liczbach naturalnych, gdzie $triangle(x, y, z)$ zachodzi gdy z odcinków o bokach długości x, y, z można zbudować trójkąt?

(c) Czy dane zdanie logiki pierwszego rzędu nad sygnaturą złożoną z jednego dwuargumentowego symbolu funkcyjnego f (i równości) jest prawdziwe w liczbach naturalnych, gdzie f jest interpretowane jako funkcja $(x, y) \mapsto (x + y)^2$?

(d) Czy dane zdanie logiki pierwszego rzędu nad sygnaturą $(+, \cdot, 0, 1)$ jest prawdziwe w ciele liczb zespolonych?

13. **Automaty i języki formalne**

Czy istnieją języki regularne $L_n \subseteq \{a, b\}^*$ o następujących własnościach? Dla każdego podpunktu podaj przykład (i uzasadnij jego poprawność) lub udowodnij, że takie języki nie istnieją.

(a) Każdy deterministyczny automat skończony rozpoznający L_n ma co najmniej $\Theta(2^n)$ stanów, ale L_n można rozpoznać niedeterministycznym automatem skończonym mającym $O(n)$ stanów.

(b) Każdy deterministyczny automat skończony rozpoznający L_n ma co najmniej $\Theta(2^n)$ stanów, ale istnieje $K_n \subseteq (\{a, b\} \times \Sigma)^*$ (dla pewnego alfabetu Σ) taki, że $L_n = \pi_1(K_n)$ oraz K_n można rozpoznać deterministycznym automatem skończonym mającym $O(n)$ stanów (operacja π_1 w każdej literze każdego słowa z K_n usuwa drugą współrzędną, pozostawiając tylko a lub b).

(c) Każdy niedeterministyczny automat skończony rozpoznający L_n ma co najmniej $\Theta(2^n)$ stanów, ale L_n^R (zbiór słów z L_n czytanych od tyłu) można rozpoznać niedeterministycznym automatem skończonym mającym $O(n)$ stanów.

(d) Każdy niedeterministyczny automat skończony rozpoznający L_n ma co najmniej $\Theta(2^n)$ stanów, ale L_n można rozpoznać niedeterministycznym automatem ze stosem mającym $O(n)$ stanów.

14. **Programowanie współbieżne i rozproszone, systemy komputerowe**

Rozważamy model komunikacji asynchronicznej:

- Wywołanie `send(p,m)` powoduje zakolejkowanie wiadomości `m` przeznaczonej dla procesu `p`. Wywołanie kończy się od razu, bez oczekiwania na dostarczenie wiadomości.
- Wywołanie `get()` przez proces p zwraca pierwszą zakolejkowaną wiadomość przeznaczoną dla p. Wiadomości dla p są odbierane w kolejności ich wysłania. Jeśli żadna wiadomość dla p nie czeka w kolejce, funkcja oczekuje na przyjście takiej wiadomości.

Rozważmy proces serwera, który zarządza dostępem do dwóch zasobów A, B. Dla każdego z nich pamięta, czy zasób jest dostępny, czy zajęty. Przechowuje także listę oczekujących żądań od innych procesów. Proces ten wykonuje w nieskończonej pętli następujące operacje:

- Odbierz wiadomość `m` funkcją `get()`.
- Jeśli `m` to `freeA` lub `freeB`, to zaznacz, że dany zasób jest dostępny.
- W przeciwnym przypadku dopisz `m` na koniec listy oczekujących żądań. Żądania są postaci (`takeA`, p) lub (`takeB`, p) i mówią, że proces p chce uzyskać dostęp do zasobu A lub B.
- W obu przypadkach, znajdź na liście oczekujących żądań pierwsze żądanie, które może zostać spełnione. Jeśli istnieje, to zaznacz, że dany zasób jest zajęty, usuń żądanie z listy, a procesowi p, który zgłosił żądanie, wyślij wiadomość `start`.

(a) Uruchamiamy wiele kopii procesu opisanego następującym pseudokodem:

```
while (true) {
  send(serw, takeA)
  send(serw, takeB)
  get()
  get()
  korzystaj_z_A_i_B()
  send(serw, freeA)
  send(serw, freeB)
  pracuj_sam()
}
```

Co może pójść źle (dlaczego napisany powyżej kod klienta może nie działać zgodnie z oczekiwaniami)?

(b) Czy można naprawić powyższy kod, nie zmieniając kodu serwera? Zakładamy, że w momencie, gdy któryś z procesów wywołuje `korzystaj_z_A_i_B`, serwer powinien wiedzieć, że oba zasoby A i B są zajęte.

(c) Załóżmy teraz, że serwer umie obsłużyć także żądanie postaci (`takeAB`, p), które mówi, że proces p chce uzyskać dostęp do obu zasobów A i B (jeśli podczas przeglądania listy oczekujących żądań serwer natrafi na takie żądanie i oba zasoby są dostępne, to zaznacza, że są zajęte, usuwa żądanie z listy i wysyła procesowi p wiadomość `start`). Uruchamiamy wiele kopii procesów trzech rodzajów, opisanych następującym pseudokodem:

```
while (true) {
  send(serw, takeA)
  get()
  korzystaj_z_A()
  send(serw, freeA)
  pracuj_sam()
}
```

```
while (true) {
  send(serw, takeB)
  get()
  korzystaj_z_B()
  send(serw, freeB)
  pracuj_sam()
}
```

```
while (true) {
  send(serw, takeAB)
  get()
  korzystaj_z_A_i_B()
  send(serw, freeA)
  send(serw, freeB)
  pracuj_sam()
}
```

Co może pójść źle (dlaczego napisany powyżej kod klienta może nie działać zgodnie z oczekiwaniami)?

(d) Czy można naprawić powyższy kod, nie zmieniając kodu serwera?

15. **Języki programowania**

(a) Wypisz jaką liczbę zwróci podana funkcja `f` w języku C dla danego wejścia `a=21, b=51`:

```
int f(int a, int b) {
      if (a > b)
          return f(a - b, b);
      if (b > a)
          return f(a, b - a);
      return a;
}
```

Napisz wersję funkcji `f` bez wywołań rekurencyjnych.

(b) Jaki będzie wynik działania poniższej klasy `Test` w języku Java? Co dokładnie wypisze na ekran? Odpowiedź uzasadnij. Opisz problemy, które tutaj dostrzegasz.

```
public class Test {
    public static void main(String[] args) {
        A red = new A("Red");
        A blue = new A("Blue");
        f(red, blue);
        System.out.println(red.getColor());
        System.out.println(blue.getColor());
        ff(blue);
        System.out.println(blue.getColor());
    }
    public static void f(Object o1, Object o2) {
        Object temp = o1;
        o1 = o2;
        o2 = temp;
    }
    private static void ff(A a) {
        a.setColor("Red");
        a = new A("Green");
        a.setColor("Yellow");
    }
}
class A {
    private String color;
    public A(String color) {
        this.color = color;
    }
    public String getColor() {
        return color;
    }
    public void setColor(String color) {
        this.color = color;
    }
}
```

(c) Opisz co robią funkcje `f1` i `f2` napisane w języku Ocaml. Jaka jest złożoność pamięciowa i czasowa działania funkcji `f1` i `f2`? Jaka jest różnica pomiędzy tymi funkcjami?

```
let rec f1 g n = if n = 1 then g else fun x -> g ((f1 g (n-1)) x);;

let f2 g n = fun x ->
    let rec h k m r = if m = 0 then r else h k (m-1) (k r) in h g n x;;
```

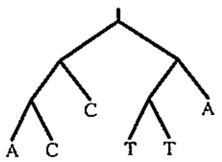
(d) Dana jest klasa implementująca drzewa w języku Python 3:

```
class Tree:
    def __init__(self):
        self.left = None
        self.right = None
        self.data = None
```

Napisz generator przechodzący drzewo dane jako argument w porządku infiksowym (in-order), czyli w porządku: lewe poddrzewo, wierzchołek, prawe poddrzewo.

16. **Bioinformatyka i biologia obliczeniowa**

(a) Przy założeniu metody tradycyjnej parsymonii podaj koszt (liczbę mutacji) i wszystkie optymalne etykietowania dla poniższego drzewa ukorzenionego, w którym na liściach znajdują się podane sekwencje jednoelementowe.



(b) Czy podaną macierz można uzupełnić do addytywnej macierzy odległości? Uzasadnij odpowiedź.

			6	
	8			
6	6	6		

(c) Czy macierz z poprzedniego punktu (16b) można uzupełnić do ultrametrycznej macierzy odległości? Uzasadnij odpowiedź.

(d) Dana jest dowolna macierz odległości M. Czy jest prawdą, że metoda UPGMA[1] zrekonstruuje z M drzewo, w którym odległości między i oraz j są równe $M[i, j]$, dla każdej pary liści i oraz j? Odpowiedź uzasadnij.

[1]Unweighted Pair Group Method with Arithmetic mean