

PESEL:

--	--	--	--	--	--	--	--	--	--

UNIwersytet Warszawski

Wydział Matematyki, Informatyki i Mechaniki

Egzamin wstępny na studia 2 stopnia na kierunku INFORMATYKA

30 czerwca 2025 roku

Czas rozwiązywania: 150 minut

*W każdym spośród 30 zadań podane są trzy warianty: (a), (b) oraz (c). W kratce przy każdym z wariantów należy odpowiedzieć, czy jest on prawdziwy, wpisując drukowanymi literami **TAK** albo **NIE**. W przypadku omyłkowego wpisu kratkę należy przekreślić i napisać jedno z tych słów po jej lewej stronie.*

Przykład poprawnego rozwiązania zadania

4. Każda liczba całkowita postaci $10^n - 1$, gdzie n jest całkowite i dodatnie,

TAK

 (a) dzieli się przez 9;

NIE

 (b) jest pierwsza;

TAK

 (c) jest nieparzysta.

*Na stronach testu można pisać wyłącznie we wskazanych wyżej miejscach i jedynie słowa **TAK** oraz **NIE**. Pisać należy długopisem lub piórem.*

Zasady punktacji

Kandydat zdobywa punkty „duże” (od 0 do 30) i punkty „małe” (od 0 do 90):

- *jeden punkt „duży” kandydat uzyskuje za zadanie, w którym poprawnie wskazał prawdziwość albo fałsz każdego z trzech związanych z tym zadaniem wariantów odpowiedzi;*
- *jeden punkt „mały” kandydat uzyskuje za każde poprawne wskazanie prawdziwości albo fałszu pojedynczego wariantu odpowiedzi. Oznacza to, że 3 „małe” punkty uzyskane w jednym zadaniu składają się na jeden „duży” punkt.*

Ostatecznym wynikiem egzaminu jest liczba

$$W = \min(30, D + m/100),$$

gdzie D oznacza liczbę „dużych”, a m liczbę „małych” punktów. Na przykład 5,50 oznacza, że kandydat poprawnie wskazał w całym teście prawdziwość albo fałsz łącznie 50 wariantów odpowiedzi, w tym każdego z trzech wariantów dla pewnych pięciu zadań.

Zasadniczą rolę w ostatecznym wyniku testu mają punkty „duże”. Punkty „małe” zwiększają rozdzielczość, jeśli wielu kandydatów dostało tyle samo „dużych” punktów.

Powodzenia!

1. Ograniczony z dołu ciąg malejący liczb rzeczywistych $\langle a_n \rangle_{n \geq 1}$

- ☐ (a) jest zbieżny;
- ☐ (b) spełnia równość $\lim_{n \rightarrow \infty} a_n = \inf_{n \geq 1} a_n$;
- ☐ (c) ma wyraz największy.

2. Szereg $\sum_{n=1}^{\infty} a_n$ o wyrazach rzeczywistych jest bezwzględnie zbieżny. Wynika z tego, że

- ☐ (a) jest on zbieżny;
- ☐ (b) szereg $\sum_{n=1}^{\infty} a_{p(n)}$ jest zbieżny dla każdej permutacji p zbioru liczb całkowitych dodatnich;
- ☐ (c) $\lim_{n \rightarrow \infty} |a_{2n}| = \lim_{n \rightarrow \infty} a_{2n+1}$.

3. Funkcja $f: [0; 7] \rightarrow \mathbb{R}$ jest różniczkowalna i $f(0) = 0$. Wynika z tego, że

- ☐ (a) $7f'(x) = f(7)$ dla pewnego $x > 0$;
- ☐ (b) f jest ciągła w 0;
- ☐ (c) f jest monotoniczna na przedziale $[0; r]$ dla pewnego $r > 0$.

4. Pewne trzy wiersze w_1, w_2, w_3 macierzy rzeczywistej A o wymiarach 25×25 tworzą układ liniowo niezależny w $\mathbb{R}^{25}$, a wszystkie pozostałe wiersze należą do przestrzeni liniowej rozpiętej przez w_1, w_2, w_3 . Wynika z tego, że

- ☐ (a) wszystkie wiersze macierzy A tworzą układ liniowo zależny w $\mathbb{R}^{25}$;
- ☐ (b) jądro macierzy A^T jest przestrzenią liniową wymiaru trzy;
- ☐ (c) istnieje złożony z trzech kolumn macierzy A układ, który jest liniowo niezależny w $\mathbb{R}^{25}$.

5. Kwadratowe macierze zespolone A, B są podobne. Wynika z tego, że

- ☐ (a) $\text{rank}(A + A^3) = \text{rank}(B + B^3)$;
- ☐ (b) $\text{im}(A + A^3) = \text{im}(B + B^3)$;
- ☐ (c) $\det(A + A^3) = \det(B + B^3)$.

6. Dodatnie liczby całkowite a, b, c, d spełniają warunek $\text{NWD}(ab, cd) = 1$. Wynika z tego, że

- ☐ (a) $\text{NWD}(abc, d) = 1$;
- ☐ (b) $\text{NWD}(a^b, c^d) = 1$;
- ☐ (c) $\text{NWW}(ac, bd) = abcd$.

7. Niech G będzie grafem dwudzielnym, w którym stopień każdego wierzchołka jest równy 2. Wynika z tego, że

- ☐ (a) graf G jest planarny;
- ☐ (b) graf G jest eulerowski;
- ☐ (c) w grafie G istnieje skojarzenie doskonałe.

8. Istnieje relacja równoważności na zbiorze $\{0, 1, 2, 3, 4, 5, 6\}$, taka że
- ☐ (a) wszystkie jej klasy abstrakcji mają taką samą liczbę elementów;
 - ☐ (b) 0 jest w relacji z 6, a nie jest w relacji z 5;
 - ☐ (c) x jest w relacji z y wtedy i tylko wtedy, gdy $x + y = 6$.
9. Porządek częściowy $(A, \leq)$ ma element najmniejszy i element największy. Wynika z tego, że
- ☐ (a) jest to porządek liniowy;
 - ☐ (b) jest to porządek dobrze ufundowany;
 - ☐ (c) pewien podzbiór $X \subseteq A$ ma kres górny i kres dolny.
10. Niech A i B będą zdarzeniami losowymi, takimi że $0 < P(A) < \frac{1}{4}$ oraz $0 < P(B) < \frac{1}{4}$. Wynika z tego, że
- ☐ (a) $P(A \cup B) < \frac{1}{2}$;
 - ☐ (b) $P(A \cap B) \leq \frac{1}{16}$;
 - ☐ (c) $P(A|B) \leq \frac{1}{4}$.
11. Rzucamy pięciokrotnie symetryczną monetą. Za orła dostajemy 1 punkt, a za reszkę dostajemy -1 punkt. Wyniki rzutów są niezależne. Niech X będzie sumą zdobytych punktów. Wynika z tego, że
- ☐ (a) $P(X = 1) > \frac{1}{4}$;
 - ☐ (b) $E(X) = 0$;
 - ☐ (c) $\text{Var}(X) = 5$.
12. W statystycznym uczeniu maszynowym błąd nieredukowalny
- ☐ (a) jest dolnym ograniczeniem błędu walidacyjnego danego modelu dla każdej próby;
 - ☐ (b) jest dolnym ograniczeniem błędu walidacyjnego danej próby dla każdego modelu;
 - ☐ (c) oznacza błąd modelu na zbiorze walidacyjnym.
13. W eksperymencie rzucamy 10 razy monetą, dla której prawdopodobieństwo wypadnięcia orła wynosi $q \in (0; 1)$. Estymujemy wartość q na podstawie liczby wyrzuconych orłów. Testujemy hipotezę $H_0: q = 0,5$, $H_1: q \neq 0,5$ testem χ^2 . Wyrzucono 3 orły. Wynika z tego, że
- ☐ (a) wartość nieobciążonego estymatora q otrzymana w tym eksperymencie to 0,3;
 - ☐ (b) $q = 0,3$;
 - ☐ (c) dla tego testu p -wartość w tym eksperymencie jest większa niż p -wartość w eksperymencie, w którym uzyskujemy 6 orłów w 20 rzutach.
14. W drzewie AVL o n wierzchołkach oznaczmy przez K długość najkrótszej, a przez D długość najdłuższej ścieżki od korzenia do liścia. Wynika z tego, że
- ☐ (a) $K + D = O(\log n)$;
 - ☐ (b) $D - K = O(1)$;
 - ☐ (c) $K \cdot D = \Omega(\log^2 n)$.

15. Tablicę $a[1..n]$ zawierającą $n > 1$ różnych liczb całkowitych nazwiemy *prawie posortowaną* wtedy i tylko wtedy, gdy do posortowania a wystarczy co najwyżej jedna zamiana miejscami pary jej elementów. Wynika z tego, że

- ☐ (a) algorytm sortowania przez wstawianie (ang. *insertion sort*) zastosowany do prawie posortowanej n -elementowej tablicy wykona w pesymistycznym przypadku $\Theta(n)$ porównań elementów w tej tablicy;
- ☐ (b) element najmniejszy w prawie posortowanej n -elementowej tablicy można znaleźć, wykonując $o(\log n)$ porównań;
- ☐ (c) średnia arytmetyczna liczby inwersji w prawie posortowanych permutacjach liczb $1, 2, \dots, 6$ wynosi co najmniej 3.

16. Niech $L_1 = \{a^k b^k c^m d^n \mid k, m, n \in \mathbb{N}\}$ oraz $L_2 = \{a^k b^m c^n d^n \mid k, m, n \in \mathbb{N}\}$. Wynika z tego, że bezkontekstowy jest język

- ☐ (a) $L_1 \cap L_2$;
- ☐ (b) $\{a, b, c, d\}^* - (L_1 \cup L_2)$;
- ☐ (c) $L_1 - L_2$.

17. Niech K, L, M będą dowolnymi językami nad alfabetem $\{a, b\}$. Wynika z tego, że

- ☐ (a) $K \cdot (L \cdot M) = (K \cdot L) \cdot M$;
- ☐ (b) $(K \cap L) \cdot M = (K \cdot M) \cap (L \cdot M)$;
- ☐ (c) $(\{a, b\}^* - L)^* \cap L^* = \emptyset$.

18. W programowaniu obiektowym pojęcie *polimorfizmu*

- ☐ (a) znajduje zastosowanie tylko i wyłącznie w języku Java;
- ☐ (b) oznacza, że obiekty różnych klas mogą reagować na różne sposoby na ten sam komunikat;
- ☐ (c) jest tożsame z pojęciem kapsułkowania.

19. Dany jest następujący program w Javie:

```
1 public class Main {
2     public static void main(String[] args) {
3         int a = 42;
4         Integer b = 42;
5         // System.out.println(a.equals(b));
6         // System.out.println(a == b++);
7         // System.out.println(b.equals(a));
8     }
9 }
```

Kompilowanie tego programu powiedzie się i program wypisze wartość `true` po odkomentowaniu linii

- ☐ (a) 5;
- ☐ (b) 6;
- ☐ (c) 7.

20. Załóżmy, że w bazie danych są tabele $R(A, B, C)$ oraz $S(A, B, D)$. Wynika z tego, że

- ☐ (a) jeśli kolumna A jest kluczem głównym w obu tabelach i dane w obu tabelach są przechowywane na dysku posortowane niemalejąco po kolumnie A , to złączenie naturalne tabel R i S można wyliczyć w czasie liniowym ze względu na sumaryczny rozmiar obu tabel;
- ☐ (b) jeśli kolumna A jest kluczem głównym w obu tabelach i w tabeli R jest m wierszy, a w tabeli S jest n wierszy, to w złączeniu naturalnym tych tabel jest co najwyżej $\min(m, n)$ wierszy;
- ☐ (c) jeśli dane w obu tabelach są przechowywane na dysku posortowane niemalejąco po kolumnie A , to złączenie naturalne tabel R i S można wyliczyć w czasie liniowym ze względu na sumaryczny rozmiar obu tabel.

21. Tabela T została zdefiniowana poleceniem

```
CREATE TABLE T (A INT NOT NULL, B INT NOT NULL, C INT, PRIMARY KEY (A, B))
```

i jej schemat nie był modyfikowany. Wynika z tego, że każde dwa wiersze tej tabeli mają

- ☐ (a) różne wartości w kolumnie A ;
- ☐ (b) różne wartości w kolumnie C ;
- ☐ (c) różne wartości w kolumnie A lub różne wartości w kolumnie B .

22. Rozważmy następujący fragment kodu HTML5:

```
<style>
  p { color: blue; }
  .highlight { color: red; }
  #intro { color: green; }
  div p { color: yellow; }
</style>
<div id="container" class="box primary">
  <p class="highlight" id="intro">This is some text.</p>
</div>
```

Element `<p class="highlight" id="intro">This is some text.</p>` ma kolor

- ☐ (a) `blue`
- ☐ (b) `red`
- ☐ (c) `green`

23. Ciasteczka HTTP stosuje się w aplikacjach WWW

- ☐ (a) przy nawiązywaniu sesji z logowaniem;
- ☐ (b) przy nawiązywaniu sesji bez logowania;
- ☐ (c) do śledzenia aktywności użytkownika w sieci.

24. Następujący fragment tablicy tras

Cel	Maska	Brama	Interfejs
10.0.0.0	x	0.0.0.0	eth0
y	255.255.240.0	0.0.0.0	eth1

jest poprawny, jeśli

- ☐ (a) x to 255.255.240.0, y zaś to 10.0.1.0;
- ☐ (b) x to 255.255.255.0, y zaś to 10.0.16.0;
- ☐ (c) x to 255.255.0.0, y zaś to 10.1.0.0.

25. Istnieje scenariusz użycia UDP i interfejsu gniazd w komunikacji przez sieć o MTU 1500, taki że proces A wysłał do procesu B dane o rozmiarze 2944 bajty za pomocą jednego wywołania funkcji `write`, a proces B odebrał

- ☐ (a) tylko początkowe 1472 bajty tych danych za pomocą jednego wywołania funkcji `read`;
- ☐ (b) całość tych danych za pomocą jednego wywołania funkcji `read`;
- ☐ (c) całość tych danych za pomocą dwóch wywołań funkcji `read`, w każdym otrzymując po 1472 bajty.

26. Balbina i Ptyś komunikują się przez internet. Balbina postępuje roztropnie, gdy chcąc

- ☐ (a) przekonać Ptysia, że jest autorką wysłanej do niego wiadomości, oblicza skrót kryptograficzny tej wiadomości i szyfruje ten skrót swoim kluczem prywatnym, a następnie wysyła mu wiadomość oraz tak zaszyfrowany jej skrót;
- ☐ (b) zapewnić, że tylko Ptyś przeczyta jej wiadomość, szyfruje tę wiadomość losowym kluczem za pomocą uzgodnionego wcześniej algorytmu symetrycznego, szyfruje ten klucz kluczem publicznym Ptysia, a następnie wysyła mu tak zaszyfrowane wiadomość i klucz;
- ☐ (c) zabezpieczyć wysyłaną do Ptysia wiadomość przed modyfikacją, oblicza jej skrót kryptograficzny i szyfruje ten skrót kluczem publicznym Ptysia, a następnie wysyła mu wiadomość oraz tak zaszyfrowany jej skrót.

27. Rozważmy dodawanie zawartości dwóch 8-bitowych rejestrów w architekturze x86. Jeśli rejestry zawierają ciągi bitów

- ☐ (a) 11111010 i 11110000, to zostanie ustawiony bit przeniesienia i wyzerowany bit nadmiaru;
- ☐ (b) 01111000 i 01101110, to zostanie wyzerowany bit przeniesienia i ustawiony bit nadmiaru;
- ☐ (c) 11100110 i 10000010, to zostaną ustawione bity przeniesienia i nadmiaru.

28. Mechanizm stronicowania stosuje się w systemach operacyjnych przy

- ☐ (a) blokowaniu procesowi dostępu do pamięci innego procesu;
- ☐ (b) zapewnianiu procesowi dostępu do pamięci innego procesu;
- ☐ (c) wykrywaniu dereferencji wskaźnika NULL przez proces.

29. Rozważmy program współbieżny składający się z dwóch przedstawionych po prawej stronie procesów współdzielących zmienną globalną x , której odczyty i zapisy są atomowe.

- ☐ (a) W dowolnym momencie jego wykonania wartość zmiennej x jest równa co najwyżej 5.
- ☐ (b) Istnieje jego wykonanie, takie że wartość zmiennej x jest stale równa 0 w trakcie i po wykonaniu obydwu procesów.
- ☐ (c) Istnieje jego wykonanie, takie że wartość zmiennej x po zakończeniu obydwu procesów jest równa 3.

```
int x = 0;

process P1() {
    int y;
    for (i = 0; i < 5; i++) {
        y = x;
        y = y + 1;
        x = y;
    }
}

process P2() {
    int y;
    for (i = 0; i < 5; i++) {
        y = x;
        y = y - 1;
        x = y;
    }
}
```

30. Rozważmy program współbieżny składający się z dwóch przedstawionych po prawej stronie procesów. Funkcja `print` wypisuje natychmiast. Wynikiem wykonania programu jest ciąg czterech cyfr.

- ☐ (a) Przy założeniu, że semafor jest semaforem Dijkstry, istnieje co najmniej pięć wykonań tego programu, takich że każde wypisuje inny wynik.
- ☐ (b) Przy założeniu, że semafor jest silnie uczciwy, istnieją co najwyżej dwa wykonania tego programu, takie że każde wypisuje inny wynik.
- ☐ (c) Liczba różnych możliwych wypisywanych wyników jest taka sama, jeśli semafor jest semaforem słabym i jeśli jest semaforem silnym.

```
semaphore S = 0;

process P1() {
    print("1");
    P(S);
    print("2");
}

process P2() {
    print("3");
    V(S);
    print("4");
}
```