# Egzamin licencjacki/inżynierski

15 lutego 2022

**Informacja dla zdających egzamin na kierunku informatyka**

Z sześciu poniższych zestawów zadań (Matematyka I, Matematyka II, Metody programowania, Matematyka dyskretna, Algorytmy i struktury danych, Metody numeryczne) należy wybrać i przedstawić na osobnych kartkach rozwiązania trzech zestawów.

**Informacja dla zdających egzamin na kierunku ISIM**

Z sześciu poniższych zestawów zadań (Matematyka I, Metody programowania, 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+30=150 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

Udowodnij, że istnieje taki trójargumentowy spójnik logiczny $\otimes$, że $\{\otimes\}$ jest zupełnym zbiorem spójników.

## Matematyka II — Algebra

Zadanie 1. (6 punktów)
Rozważamy przestrzeń $\Pi_4$ wielomianów stopnia $\leq 4$.
a) (2p.) Wykazać, że wielomiany $p_i(x) = \Pi_{j=0}^{i-1}(x-j)$, $(i = 1, \ldots, 5)$ tworzą bazę tej przestrzeni.
b) (4p.) Znaleźć współczynniki rozwinięcia wielomianu $w(x) = x^4 - 7x^2 + 16x^2 - 12x + 3$ względem tej bazy. (Rozwiązanie korzystające z postaci potęgowej to 2 pkt.)

Zadanie 2. (8 punktów)
Rozważamy przestrzeń liniową $\mathbb{R}^3$ i przekształcenie $\tau : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$, określone jako $\tau(x, y) = x_1y_1 + 2x_2y_2 + x_3y_3$.
a) (2p.) Jakie warunki powinno spełniać przekształcenie $\tau$ aby było iloczynem skalarnym (wymienić, nie dowodzić).
b) (2p.) Obliczyć długość wektora $x = [1\ 2\ 1]^T$.
c) (4p.) Uzupełnić wektor $x$ do bazy ortogonalnej przestrzeni $\mathbb{R}^3$ (ortogonalność względem iloczynu skalarnego $\tau$).

Progi punktowe: 4, 6, 8, 10, 12 punktów.

## Metody Programowania

Poniższe zadania należy rozwiązać używając języka funkcyjnego. Do wyboru: Racket, OCaml i Haskell.

**Zadanie 1** Zaimplementuj funkcję, która przyjmuje pojedynczy argument – listę długości $n$ parami różnych znaków (typu `char` w językach Racket i OCaml czy `Char` w Haskellu) i zwraca listę wszytkich słów długości $n$, reprezentowanych jako listy znaków, w których znaki się nie powtarzają i które można ułożyć ze znaków podanych w argumencie.

**Zadanie 2** Zaproponuj reprezentację drzew o etykietach zarówno w węzłach wewnętrznych jak i w liściach, w których węzeł może mieć dowolnie dużo dzieci. Możesz ustalić typ etykiet lub uczynić go parametrem swojej definicji. Jak wygląda zasada indukcji strukturalnej dla Twojego typu danych?

**Zadanie 3** Dla swojej reprezentacji drzew z zadania 2 napisz funkcję `flatten`, która dla danego drzewa zwraca listę wszystkich jego etykiet czytanych w porządku *preorder*.

**Uwaga!** Jakość kodu, a zwłaszcza struktura definicji, ma znaczenie i będzie punktowana.

## Matematyka dyskretna

Niech $x_k = 111\ldots111$ będzie liczbą, której zapis w układzie dziesiętnym składa się z $k$ jedynek. Niech $P(m)$ jest prawdą wtedy i tylko wtedy gdy dla nieskończenie wielu liczb $k$ zachodzi $m|x_k$. Scharakteryzuj liczby $m$, dla których zachodzi $P(m)$.

## 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. $\boxed{\textbf{4 punkty}}$ Niech dana będzie liczba naturalna $n$. W języku programowania `PWO++` procedura `Chebyshev2Power(`$n$`)` znajduje taki wektor liczb rzeczywistych $[a_0^{(n)}, a_1^{(n)}, \ldots, a_n^{(n)}]$, że
$$T_n(x) = \sum_{k=0}^{n} a_k^{(n)} \cdot x^k,$$
gdzie $T_n$ oznacza $n$-ty wielomian Czebyszewa. Krótko mówiąc: procedura ta wyznacza współczynniki postaci potęgowej wielomianu Czebyszewa stopnia $n$. **Niestety ma ona pewną wadę**, a mianowicie – stopień $n$ nie może być większy niż 2021. Wykorzystując procedurę `Chebyshev2Power`, zaproponuj *efektywną* metodę znajdowania postaci potęgowej wielomianu Czebyszewa $T_{2023}$, tj. obliczania współczynników $a_k^{(2023)}$ $(0 \le k \le 2023)$.

2. $\boxed{\textbf{4 punkty}}$ Przedstaw ideę kwadratur złożonych. Wyprowadź złożony wzór trapezów.

3. $\boxed{\textbf{4 punkty}}$ Opisz metodę faktoryzacji rozwiązywania nieosobliwego układu równań liniowych. Określ jej złożoność czasową i pamięciową.

## 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: sortowanie leksykograficzne ciągów o takiej samej długości (4 punkty)

Rozważmy następującą wersję problemu sortowania leksykograficznego. Danych jest $n$ ciągów $s_1$, $s_2$, $\ldots$, $s_n$ zbudowanych nad $m$-literowym alfabetem $\Sigma$ (można założyć, że $m$ jest pewną niewielką stałą); wszystkie ciągi mają tą samą długośc $k$. Zadanie polega na uporządkowaniu leksykograficznym tych ciągów.

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

### Zadanie 2: kopiec minimaksowy (5 punktów)

Zaprojektuj strukturę takiej kolejki priorytetowej, z której będzie można pobierać zarówno element minimalny jak i maksymalny. Twoje rozwiązanie powinno opierać się na kopcu binarnym a złożoność czasowa operacji *insert(x)*, *extract–min()* i *extract–max()* powinna być logarytmiczna względem aktualnej liczby elementów w strukturze.

Opisz krótko ideę działania tej struktury a następnie zapisz w pseudokodzie procedury realizujące wstawienie elementu do kolejki oraz pobranie elementu minimalnego i maksymalnego z kolejki. Uzasadnij, że przedstawiona implementacja operacji kolejkowych będzia działać w czasie logarytmicznym.

## Języki formalne i złożoność obliczeniowa

Problem Wszyscy-Na-Kwarantannę jest zdefiniowany następująco:
**Wejście**. Wejście składa się z liczby `limit`, listy nazwisk studentów `students` oraz listy grup `groups`, zawierającej listy nazwisk uczestników poszczególnych grup (oraz, być może, inne dane). Przykładowo, w formacie JSON wejście mogłoby wyglądać tak:

```
{
   "limit":"2",
   "students":["Kowalski","Morawiecki","Miodek","Sutryk","Pacholski"],
   "groups":[
      {
         "group_id": "18353",
         "day": "Monday",
         "name": "Logika dla informatyków",
         "participants":["Kowalski","Morawiecki","Miodek"]
      },
      {
         "group_id": "18364",
         "day": "Monday",
         "name": "Języki formalne i nieformalne",
         "participants":["Sutryk","Miodek"]
      }
   ]
}
```

**Wyjście**. `TAK`, jeśli można wybrać tylu studentów, ile wynosi `limit` tak, aby każdy student spośród `students` był albo wybrany, albo był w jakiejś grupie z wybraną osobą; inaczej `NIE`.

### Zadanie

Jaką złożoność ma problem Wszyscy-Na-Kwarantannę?

### Warsztat

Wolno korzystać z następujących faktów: problemy 3SAT, SAT, NAESAT, 3COL, Clique, Vertex-Cover, Subset-sum, Hamilton i TSP są NP-zupełne, problem TQBF jest PSPACE-zupełny, P $\neq$ EXPTIME, PSPACE $\neq$ EXPSPACE oraz PSPACE=NPSPACE. Pozostałe fakty, z których chce się skorzystać, należy dowieść.