SPRAWDZIAN · INFORMATYKA

Sprawdzian z teorii informatyki **Grupa A**

Informatyka: budowa komputera, systemy operacyjne, sieci, bezpieczeństwo i obliczalność

Imię i nazwisko: _____

Klasa: _____

Data: _____

Punkty: _____ / 25

10 zadań, 25 pkt. Czytaj uważnie polecenia. W zadaniach otwartych zapisz obliczenia.

Zadanie 1. (0–1 pkt)

Który protokół warstwy transportowej modelu OSI/TCP gwarantuje niezawodne dostarczenie danych z potwierdzeniami odbioru i kontrolą przepływu?

(A) UDP
(B) TCP
(C) ICMP
(D) ARP

Zaznacz jedną poprawną odpowiedź.

Zadanie 2. (0–1 pkt)

Program typu ransomware:

(A) Monitoruje naciskane klawisze
(B) Wyświetla niechciane reklamy
(C) Kradnie dane logowania z przeglądarki
(D) Szyfruje pliki użytkownika i żąda okupu

Zaznacz jedną poprawną odpowiedź.

Zadanie 3. (0–3 pkt)

Oceń prawdziwość stwierdzeń dotyczących złożoności obliczeniowej:

1. Dla problemu komiwojażera (TSP) nie jest znany żaden algorytm działający w czasie wielomianowym.	**P**	**F**
2. Klasa P zawiera problemy decyzyjne rozwiązywalne w czasie wykładniczym, ale nie wielomianowym, na deterministycznej maszynie Turinga.	**P**	**F**
3. Gdyby udowodniono P = NP, każdy problem z klasy NP dałoby się rozwiązać w czasie wielomianowym.	**P**	**F**

Przy każdym zdaniu zakreśl P (prawda) albo F (fałsz).

Zadanie 4. (0–3 pkt)

Oceń prawdziwość stwierdzeń dotyczących teorii automatów i złożoności:

1. Klasa P jest podzbiorem klasy NP.	**P**	**F**
2. Automat skończony deterministyczny (DFA) jest mniej potężny niż niedeterministyczny (NFA) i nie może rozpoznać niektórych języków regularnych.	**P**	**F**
3. Każdy język bezkontekstowy jest również językiem regularnym.	**P**	**F**

Przy każdym zdaniu zakreśl P (prawda) albo F (fałsz).

Zadanie 5. (0–4 pkt)

Dopasuj warstwę modelu ISO/OSI do jej głównego zadania.

1. Warstwa fizyczna	…	A. Przesyłanie surowych bitów przez medium transmisyjne
2. Warstwa łącza danych	…	B. Zapewnienie niezawodnego dostarczania danych między aplikacjami (np. protokół TCP)
3. Warstwa sieciowa	…	C. Trasowanie (routing) pakietów między różnymi sieciami na podstawie adresów IP
4. Warstwa transportowa	…	D. Adresowanie fizyczne (MAC) i wykrywanie błędów transmisji w obrębie jednego segmentu sieci

Przy każdym numerze wpisz literę pasującego elementu.

Zadanie 6. (0–2 pkt)

Uzupełnij zdania dotyczące sieci komputerowych: (1) System tłumaczący nazwy domenowe (np. www.matury-online.pl) na adresy IP nazywa się (1) _____ . (2) Urządzenie lub oprogramowanie kontrolujące ruch sieciowy i blokujące niepożądane połączenia zgodnie z zadanymi regułami to (2) _____ .

Zadanie 7. (0–2 pkt)

Wyjaśnij różnicę między procesem a wątkiem (thread) w systemie operacyjnym. Dlaczego przełączanie między wątkami tego samego procesu jest zazwyczaj szybsze niż przełączanie między procesami?

Zadanie 8. (0–3 pkt)

Opisz problem zatrzymania (halting problem). Dlaczego jest nierozstrzygalny?

Zadanie 9. (0–3 pkt)

Wyjaśnij czym jest gramatyka bezkontekstowa (CFG). Podaj jej formalne elementy składowe oraz przykład reguł produkcji opisujących proste wyrażenia arytmetyczne (np. $a + b \times c$).

Zadanie 10. (0–3 pkt)

Wyjaśnij, czym jest problem NP-zupełny. Podaj przykład takiego problemu i wyjaśnij, dlaczego dowód P=NP miałby fundamentalne znaczenie dla informatyki i kryptografii.

Rozwiąż ten sprawdzian online, z wyjaśnieniem każdej odpowiedzi: matury-online.pl/sprawdzian/informatyka/teoria-informatyki

Sprawdzian z teorii informatyki **Grupa B**

Informatyka: budowa komputera, systemy operacyjne, sieci, bezpieczeństwo i obliczalność

Imię i nazwisko: _____
Klasa: _____
Data: _____
Punkty: _____ / 25

10 zadań, 25 pkt. Czytaj uważnie polecenia. W zadaniach otwartych zapisz obliczenia.

Zadanie 1. (0–1 pkt)

Która klasa złożoności obejmuje problemy rozwiązywalne deterministycznie w czasie wielomianowym?

(A) EXPTIME
(B) NP
(C) PSPACE
(D) P

Zaznacz jedną poprawną odpowiedź.

Zadanie 2. (0–1 pkt)

W modelu OSI warstwa transportowa (Transport Layer, warstwa 4) odpowiada za:

(A) Konwersję sygnałów elektrycznych i fizyczną transmisję bitów
(B) Wyznaczanie optymalnych ścieżek pakietów między różnymi sieciami
(C) Niezawodny transfer danych end-to-end między procesami na hostach
(D) Formatowanie i szyfrowanie danych dla aplikacji użytkownika

Zaznacz jedną poprawną odpowiedź.

Zadanie 3. (0–3 pkt)

Oceń prawdziwość stwierdzeń o chmurze obliczeniowej (cloud computing):

1. Model chmurowy pozwala na elastyczne skalowanie zasobów w zależności od potrzeb.	**P**	**F**
2. SaaS (Software as a Service) wymaga instalacji oprogramowania na komputerze użytkownika.	**P**	**F**
3. IaaS (Infrastructure as a Service) udostępnia wirtualne maszyny i infrastrukturę.	**P**	**F**

Przy każdym zdaniu zakreśl P (prawda) albo F (fałsz).

Zadanie 4. (0–3 pkt)

Oceń prawdziwość stwierdzeń o wirtualizacji:

1. Konteneryzacja (np. Docker) wymaga osobnego jądra systemu dla każdego kontenera.	**P**	**F**
2. Maszyna wirtualna (VM) symuluje kompletny komputer z własnym systemem operacyjnym.	**P**	**F**
3. Hypervisor typu 1 (bare-metal) działa jako zwykła aplikacja w systemie operacyjnym hosta.	**P**	**F**

Przy każdym zdaniu zakreśl P (prawda) albo F (fałsz).

Zadanie 5. (0–4 pkt)

Uporządkuj pamięci od najszybszej do najwolniejszej, dopasowując do pozycji:

1. Pozycja 4 (najwolniejsza)	...	A. Pamięć RAM
2. Pozycja 2	...	B. Dysk SSD
3. Pozycja 3	...	C. Rejestry procesora
4. Pozycja 1 (najszybsza)	...	D. Pamięć cache L1

Przy każdym numerze wpisz literę pasującego elementu.

Zadanie 6. (0–2 pkt)

Uzupełnij zdania z zakresu teorii informatyki: (1) Klasa problemów rozwiązywalnych w czasie wielomianowym przez deterministyczną maszynę Turinga nazywana jest klasą (1) _____ (2) Program tłumaczący cały kod źródłowy na kod maszynowy przed jego uruchomieniem (a nie w trakcie wykonywania) nazywa się (2) _____

Zadanie 7. (0–2 pkt)

Wyjaśnij, czym jest wirtualizacja w informatyce, oraz jaką korzyść daje uruchamianie wielu maszyn wirtualnych na jednym fizycznym serwerze.

Zadanie 8. (0–3 pkt)

Wyjaśnij różnicę między kompilacją a interpretacją kodu źródłowego programu. Podaj przykład języka programowania wykorzystującego każde z tych podejść.

Zadanie 9. (0–3 pkt)

Wyjaśnij różnicę między złożonością czasową pesymistyczną (worst-case) a średnią (average-case) algorytmu. Podaj przykład algorytmu, dla którego te dwie miary istotnie się różnią.

Zadanie 10. (0–3 pkt)

Opisz czym jest deterministyczny automat skończony (DFA). Podaj elementy formalnej definicji automatu i opisz język, który akceptuje automat z jednym stanem początkowym, jednym stanem akceptującym i przejściem na symbol '1'.

Rozwiąż ten sprawdzian online, z wyjaśnieniem każdej odpowiedzi: matury-online.pl/sprawdzian/informatyka/teoria-informatyki

KLUCZ ODPOWIEDZI · DLA NAUCZYCIELA

Sprawdzian z teorii informatyki

Grupa A

1. (1 pkt) **B**

2. (1 pkt) **D**

3. (3 pkt) **1. P; 2. F; 3. P** — po 1 pkt za zdanie

4. (3 pkt) **1. P; 2. F; 3. F** — po 1 pkt za zdanie

5. (4 pkt) **1–A, 2–D, 3–C, 4–B** — po 1 pkt za parę

6. (2 pkt) **(1) DNS; (2) zapora sieciowa** — po 1 pkt za lukę

7. (2 pkt) Przykładowa odpowiedź: Proces to samodzielnie wykonywany program posiadający własną, odizolowaną przestrzeń adresową pamięci, natomiast wątek to jednostka wykonania działająca wewnątrz procesu — jeden proces może mieć wiele wątków, które współdzielą tę samą pamięć i zasoby procesu. Przełączanie kontekstu między wątkami tego samego procesu jest szybsze niż między procesami, ponieważ system operacyjny nie musi zmieniać mapowania pamięci (przestrzeni adresowej) — wątki już współdzielą te same dane i kod, więc trzeba przełączyć jedynie stan rejestrów procesora, a nie całą tablicę stron pamięci.
Punktacja: Poprawne rozróżnienie procesu (samodzielny program z własną przestrzenią adresową) od wątku (jednostka wykonania wewnątrz procesu) (1 pkt) + wyjaśnienie, że wątki tego samego procesu współdzielą pamięć, więc przełączanie między nimi nie wymaga zmiany kontekstu pamięci, w przeciwieństwie do procesów (1 pkt). 2 pkt.

8. (3 pkt) Przykładowa odpowiedź: Halting problem: czy istnieje algorytm HALT(P,W) który odpowie, czy P zatrzyma się na W? Turing (1936) udowodnił nierozstrzygalność przez sprzeczność: gdyby HALT istniało, można zbudować program D który się zapętla gdy HALT(D,D)=tak i zatrzymuje gdy HALT(D,D)=nie - sprzeczność. Wniosek: niektórych problemów nie można rozwiązać algorytmicznie.
Punktacja: 1. Opis: czy program P zatrzyma się dla wejścia W? (1 pkt). 2. Dowód: sprzeczność przez auto-referencję (1 pkt). 3. Konsekwencje: istnieją problemy poza MT (1 pkt).

9. (3 pkt) Przykładowa odpowiedź: CFG to czwórka (V, Σ, R, S): V — skończony zbiór nieterminali (np. E, T, F), Σ — alfabet terminali (np. +, ×, id, (,)), R — zbiór reguł produkcji postaci $A \to \alpha$, S — symbol startowy. Reguły dla wyrażeń: $E \to E + T \mid T$; $T \to T \times F \mid F$; $F \to \mathrm{id} \mid (E)$. Zastosowanie: parsery i kompilatory używają CFG do opisu gramatyki języków programowania.
Punktacja: Formalna definicja CFG jako czwórki (V, Σ, R, S): nieterminale V, terminale Σ, reguły R, symbol startowy S — 1 pkt. Poprawne reguły produkcji (np. $E \to E + T \mid T$, $T \to T \times F \mid F$, $F \to \mathrm{id} \mid (E)$) lub uproszczone poprawne reguły — 1 pkt. Zastosowanie (parsery, kompilatory, opis składni języków programowania) — 1 pkt.

10. (3 pkt) Przykładowa odpowiedź: Problem NP-zupełny to problem, który należy do klasy NP (rozwiązanie można zweryfikować w czasie wielomianowym) i jest NP-trudny (każdy problem z NP można do niego wielomianowo zredukować). Przykład: problem spełnialności formuł logicznych (SAT) — czy istnieje przypisanie wartości logicznych spełniające formułę. Dowód P=NP oznaczyłby, że wszystkie problemy NP-zupełne mają wydajne algorytmy: można by lamic szyfry RSA i AES w czasie wielomianowym, co załamałoby bezpieczeństwo internetu, ale rownoczesnie umożliwiłoby efektywne leczenie optimalizacji trasy dostaw, foldowania białek i tysięcy innych zadań.
Punktacja: Kryterium 1 (1pkt): Poprawna definicja NP-zupełności — problem należy do NP (weryfikowalny wielomianowo) i każdy problem z NP można do niego wielomianowo zredukować. Kryterium 2 (1pkt): Poprawny przykład: SAT, problem komiwojażera (TSP), kolorowanie grafu, problem plecakowy, pokrycie wierzchołkowe. Kryterium 3 (1pkt): Implikacje P=NP — efektywne łamanie szyfrów RSA/AES, rozwiązanie problemów optymalizacyjnych, rewolucja w bioinformatyce i logistyce. 3 pkt.

Grupa B

1. (1 pkt) **D**

2. (1 pkt) **C**

3. (3 pkt) **1. P; 2. F; 3. P** — po 1 pkt za zdanie

4. (3 pkt) **1. F; 2. P; 3. F** — po 1 pkt za zdanie

5. (4 pkt) **1–B, 2–D, 3–A, 4–C** — po 1 pkt za parę

6. (2 pkt) **(1) P; (2) kompilator** — po 1 pkt za lukę

7. (2 pkt) Przykładowa odpowiedź: Wirtualizacja polega na tworzeniu programowej (wirtualnej) wersji zasobów sprzętowych, np. całego komputera z własnym systemem operacyjnym, działającej w izolowanym środowisku na fizycznym serwerze, nadzorowanej przez specjalne oprogramowanie zwane hipernadzorcą (hypervisorem). Uruchamianie wielu maszyn wirtualnych na jednym serwerze pozwala efektywniej wykorzystać jego zasoby (procesor, pamięć), zamiast utrzymywać osobny fizyczny serwer dla każdej aplikacji, a przy tym izoluje środowiska od siebie, dzięki czemu awaria lub błąd w jednej maszynie wirtualnej nie wpływa na pozostałe.
Punktacja: Poprawne wyjaśnienie pojęcia: technika tworzenia wirtualnych (symulowanych programowo) zasobów sprzętowych, np. wirtualnych komputerów, działających niezależnie od siebie na jednym fizycznym urządzeniu, nadzorowanych przez hipernadzorcę (hypervisor) (1 pkt). Poprawne wskazanie korzyści, np. lepsze wykorzystanie zasobów sprzętowych jednego serwera, izolacja aplikacji/środowisk, oszczędność kosztów sprzętu (1 pkt). Łącznie 2 pkt.

8. (3 pkt) Przykładowa odpowiedź: Kompilacja polega na przetłumaczeniu całego kodu źródłowego na kod maszynowy przed uruchomieniem programu — powstaje samodzielny plik wykonywalny (np. .exe), który można uruchamiać wielokrotnie bez ponownego tłumaczenia. Przykładem języka kompilowanego jest C++. Interpretacja polega na odczytywaniu i wykonywaniu kodu źródłowego instrukcja po instrukcji w trakcie działania programu, bez tworzenia osobnego pliku wykonywalnego — wymaga obecności interpretera przy każdym uruchomieniu. Przykładem języka interpretowanego jest Python.
Punktacja: 1 pkt — poprawny opis kompilacji (cały kod źródłowy jest tłumaczony na kod maszynowy przed uruchomieniem, powstaje plik wykonywalny). 1 pkt — poprawny opis interpretacji (kod jest odczytywany i wykonywany instrukcja po instrukcji przez interpreter, bez wcześniejszego tworzenia pliku wykonywalnego). 1 pkt — poprawne przykłady języków (np. C/C++ jako kompilowany, Python jako interpretowany). Łącznie 3 pkt.

9. (3 pkt) Przykładowa odpowiedź: Złożoność pesymistyczna (worst-case) określa maksymalną liczbę operacji, jaką algorytm może wykonać dla najgorszego możliwego zestawu danych wejściowych o danym rozmiarze. Złożoność średnia (average-case) opisuje oczekiwaną liczbę operacji przy założeniu typowego, losowego rozkładu danych wejściowych. Przykładem jest algorytm quicksort — jego złożoność pesymistyczna wynosi $O(n^2)$ i występuje np. dla danych już posortowanych przy naiwnym wyborze pivotu, natomiast złożoność średnia wynosi O(n log n) i dotyczy typowych, losowo ułożonych danych.
Punktacja: Poprawne wyjaśnienie złożoności pesymistycznej jako maksymalnej liczby operacji dla najgorszych możliwych danych wejściowych (1 pkt) + poprawne wyjaśnienie złożoności średniej jako oczekiwanej liczby operacji przy losowym rozkładzie danych (1 pkt) + trafny przykład, np. quicksort: pesymistyczna $O(n^2)$, średnia O(n log n) (1 pkt). Razem 3 pkt.

10. (3 pkt) Przykładowa odpowiedź: DFA to krotka (Q, Sigma, delta, q0, F) gdzie Q - stany, Sigma - alfabet, delta - funkcja przejścia, q0 - stan początkowy, F - stany akceptujące. Automat z jednym stanem q0 (akceptującym) i przejściem delta(q0, '1') = q0 akceptuje język 1* (dowolna liczba jedynek, łącznie z pustym słowem).
Punktacja: 1. Definicja DFA: 5-elementowa krotka (Q, Sigma, delta, q0, F) (1 pkt). 2. Opis elementów: stany, alfabet, funkcja przejścia, stan początkowy, stany akceptujące (1 pkt). 3. Przykład: automat akceptujący ciągi złożone wyłącznie z '1' (ciągi ze zbioru 1*) (1 pkt).

Skala ocen (orientacyjna, 25 pkt w każdej grupie)

bardzo dobry (5)	23–25 pkt
dobry (4)	19–22 pkt
dostateczny (3)	13–18 pkt
dopuszczający (2)	8–12 pkt
niedostateczny (1)	0–7 pkt

Progi: 90%, 75%, 50%, 30%. Progi w Twojej szkole mogą być inne. W zadaniach otwartych uznaj każdą odpowiedź poprawną merytorycznie, także sformułowaną inaczej niż przykład.

Ten sprawdzian online, z wyjaśnieniem każdej odpowiedzi: **matury-online.pl/sprawdzian/informatyka/teoria-informatyki**. Nauczyciele i korepetytorzy ułożą własną kartę z banku ponad 20 tys. zadań: **matury-online.pl/dla-korepetytorow**.