

**Egzamin licencjacki na kierunku
INFORMATYKA
w Instytucie Informatyki i Matematyki Komputerowej
(od roku akademickiego 2022/23)**

ZASADY PRZEPROWADZANIA EGZAMINU

1. Zadania na egzaminie mają na celu sprawdzenie podstawowej wiedzy i umiejętności związanych z materiałem przedstawionym na studiach pierwszego stopnia.
2. Egzamin ma formę pisemną, składa się z od 10 do 15 pytań. Szczegółowy zakres materiału opisuje katalog poniżej, podzielony na cztery działy. Zestaw pytań egzaminacyjnych dotyczy zagadnień wymienionych we wszystkich działach (przynajmniej po dwa zagadnienia z każdego działu).
3. Jedno konkretne pytanie egzaminacyjne może łączyć w sobie wiele zagadnień z katalogu (lub ich części). Na przykład umiejętność dowodzenia twierdzeń przy użyciu zasady indukcji matematycznej (zagadnienie 1) może być wymagana przy rozwiązaniu innego problemu, np. dotyczącego obliczenia złożoności algorytmu (zagadnienie 33). Jeśli w zagadnieniu jest mowa o "podanym przykładzie", zostanie on opisany w treści zadania.
4. Do zdania egzaminu wymagane jest zdobycie więcej niż 50% całkowitej liczby punktów.
5. Terminy egzaminów licencjackich ustalane są przez Dyрекcję IiMK; pierwszy termin egzaminu odbywa się końcem czerwca/początkiem lipca (szczegółowe informacje zostaną podane w terminie późniejszym).
6. Osoby, które nie zdały egzaminu w pierwszym terminie (lub do niego nie podeszły) mogą go poprawiać w drugim terminie.

Zagadnienia

Matematyczne podstawy informatyki

1. Udowodnić podane twierdzenie za pomocą zasady indukcji matematycznej.
2. Udowodnić podane twierdzenie metodą nie wprost.
3. Sprawdzić czy podana relacja jest porządkiem częściowym lub liniowym.
4. Wyznaczyć elementy największe, najmniejsze, maksymalne i minimalne dla podanego porządku.
5. Sprawdzić czy podana relacja jest relacją równoważności; wyznaczyć jej zbiór ilorazowy.
6. zilustrować na podanym przykładzie działanie metody siecznych i metody Newtona (stycznych).
7. Rozwiązać podany układ równań liniowych metodą eliminacji Gaussa.
8. Wykonać kilka iteracji metody Gaussa-Seidla na podanym przykładzie.
9. Wyznaczyć dla podanej macierzy przybliżenie największej/najmniejszej wartości własnej oraz odpowiadającego jej wektora własnego używając metody potęgowej.
10. Dla podanego zestawu danych znaleźć wielomian interpolacyjny (w postaci Lagrange'a i/lub Newtona).
11. Podać przykłady zastosowania zmiennych losowych dyskretnych i ich najważniejsze rozkłady.
12. Podać przykłady zastosowania zmiennych losowych ciągłych i ich najważniejsze rozkłady.
13. Omówić wykorzystanie łańcuchów Markowa do modelowania; obliczyć rozkład stacjonarny dla podanego łańcucha.
14. Dobrać test statystyczny do zadanego problemu.
15. Zinterpretować wyniki podanego testu statystycznego.
16. Zastosować wzór Bayesa i podać jego interpretację w danym przykładzie.
17. Sprawdzić ortogonalność / liniową niezależność podanego zestawu wektorów w przestrzeni \mathbb{R}^n .
18. Przeprowadzić ortonormalizację Grama-Schmidta podanego zestawu wektorów.
19. Omówić zastosowania rozszerzonego algorytmu Euklidesa w arytmetyce modularnej.
20. Obliczyć zadaną liczbę Stirlinga I / II rodzaju i podać jej interpretację.
21. Wykonać podane obliczenia stosując twierdzenia: Eulera, Fermata i/lub chińskie o resztach.
22. Sprawdzić istnienie konfiguracji i t -konfiguracji kombinatorycznej o zadanych parametrach; omówić zastosowanie konfiguracji do projektowania eksperymentów.

23. Wyznaczyć cykl Hamiltona, obwód Eulera i liczbę chromatyczną dla podanego grafu.
24. Znaleźć maksymalny przepływ i minimalny przekrój w danej sieci metodą Forda-Fulkersona.
25. Rozwiązać podane równanie rekurencyjne przy użyciu funkcji tworzących lub równania charakterystycznego.
26. Obliczyć granicę podanego ciągu liczbowego.
27. Obliczyć granicę podanej funkcji w punkcie; sprawdzić ciągłość funkcji.
28. Obliczyć pochodną podanej funkcji jednej zmiennej; obliczyć pochodne cząstkowe funkcji wielu zmiennych.
29. Wyznaczyć ekstrema lokalne podanej funkcji jednej / wielu zmiennych.
30. Obliczyć całkę oznaczoną podanej funkcji jednej / wielu zmiennych.

Teoretyczne podstawy informatyki

31. Udowodnić formalnie poprawność podanego algorytmu zgodnie z założeniami.
32. Sprawdzić rozstrzygalność lub częściową rozstrzygalność podanego języka, za pomocą odpowiedniej modyfikacji maszyny Turinga.
33. Wyznaczyć złożoność optymistyczną, średnią i pesymistyczną podanego algorytmu.
34. Wykorzystać odpowiednią implementację listy w podanym algorytmie tak, aby osiągnąć zadaną złożoność.
35. Zastosować odpowiednią implementację kolejki priorytetowej w celu zapewnienia zadanej złożoności podanego algorytmu.
36. Przedstawić działanie algorytmu QuickSort; podać sposoby wyboru elementu dzielącego i określić jak one wpływają na złożoność tego algorytmu w zależności od charakteru danych wejściowych.
37. Przedstawić działanie algorytmu MergeSort w wersji rekurencyjnej i iteracyjnej.
38. Wykonać kilka kroków sortowania przez zliczanie, sortowania kubełkowego oraz sortowania pozycyjnego. Porównać złożoności czasowe tych algorytmów z wybranym sortowaniem przez porównania.
39. Zilustrować działanie algorytmu Hoare'a na podanym przykładzie (problem selekcji).
40. Zilustrować na podanym przykładzie działanie algorytmu wyznaczania spójnych składowych grafu nieskierowanego.
41. Przedstawić definicję kopca typu MAX i jego reprezentację tablicową oraz zilustrować na zadanym przykładzie działanie operacji upheap, downheap.
42. Zilustrować na podanym przykładzie działanie algorytmu odtwarzania drzewa binarnego przy zadanych listach węzłów w porządkach preorder, inorder i postorder.

43. Podać sposób reprezentacji wyrażeń arytmetycznych przy pomocy drzew binarnych i zastosować go do konwersji podanego wyrażenia z postaci ONP do infiksowej.
44. zilustrować na podanym przykładzie drzewa BST sposób działania oraz złożoność operacji wyszukiwania, usuwania i dodawania węzłów.
45. Wykorzystując podany przykład wykonać operacje dodawania, usuwania wierzchołków w zadanym typie B-drzewa; określić złożoność tych operacji.
46. Przedstawić na podanym przykładzie sposób działania operacji dodawania i usuwania węzłów w drzewie AVL; określić złożoność tych operacji.
47. Omówić sposoby reprezentacji i przeglądania grafów; w zależności od wymaganej w zadaniu złożoności wykorzystać odpowiedni (wszerz lub w głąb) sposób przeglądania grafu.
48. Wyznaczyć najkrótszą ścieżkę w podanym grafie za pomocą algorytmu Dijkstry lub Bellmana-Forda.
49. Zaprezentować działanie algorytmów wyznaczania minimalnego drzewa rozpinającego (Boruvki, Prima i Kruskala) na podanym przykładzie. Podać złożoności tych algorytmów.
50. Wyznaczyć otoczkę wypukłą podanego zbioru punktów za pomocą algorytmu Grahama, Jarvisa lub przyrostowego (quickhull).
51. Sprawdzić przynależność danego problemu do klas P, NP i NP-zupełnej.
52. Zminimalizować podany automat z wykorzystaniem prawej kongruencji automatowej.
53. Wyznaczyć automat minimalny dla zadanego języka z wykorzystaniem prawej kongruencji syntaktycznej.
54. Wykorzystać lemat o pompowaniu do pokazania, iż podany język nie jest regularny.
55. Skonstruować wyrażenie regularne dla języka regularnego podanego w sposób opisowy.
56. Wyznaczyć gramatykę bezkontekstową dla zadanego języka bezkontekstowego.
57. Sprawdzić, wykorzystując algorytm CYK, czy podane słowo należy do języka generowanego przez zadaną gramatykę bezkontekstową.
58. Przedstawić obliczenia wykonywane przez zadany automat ze stosem lub maszynę Turinga w postaci ciągu opisów chwilowych.

Wytwarzanie oprogramowania

59. Wykonać operacje arytmetyczne (+, -) na podanych liczbach całkowitych w zadanej reprezentacji.
60. Wykonać operacje arytmetyczne (+, -, *, /) na podanych liczbach rzeczywistych w wybranej reprezentacji zmiennoprzecinkowej.
61. Wyjaśnić zastosowanie funkcji statycznych, pól statycznych i funkcji wirtualnych na wskazanych przykładach.

62. Omówić na podanych przykładach różne sposoby przekazywania argumentów do funkcji.
63. Przeanalizować wykorzystanie wskaźników i referencji w podanych przykładach kodu w C++.
64. Zastosować podejście obiektowe (klasy, dziedziczenie, abstrakcja, enkapsulacja i polimorfizm) do rozwiązywania podanego problemu.
65. Wyjaśnić mechanizm przeciążania operatorów i wykorzystania (w tym kontekście) funkcji zaprzyjaźnionych w C++.
66. Wykorzystać szablony klas i metod w podanych przykładach kodu w języku C++.
67. Omówić sposób obsługi plików w języku C (FILE, EOF; funkcje fopen, fclose, fprintf, fscanf).
68. Przyporządkować podane wymagania do odpowiedniej grupy (podział FURPS).
69. Zdefiniować i opisać rolę obiektów encji, brzegowych i sterujących w modelowaniu obiektowym, a także sposób ich użycia w diagramie sekwencji.
70. Omówić zadany wzorzec z grupy wzorców kreacyjnych (Singleton, Fabryka, Budowniczy) i podać przykład jego użycia.
71. Omówić zadany wzorzec z grupy wzorców strukturalnych (Adapter, Proxy, Fada) i podać przykład jego użycia.
72. Omówić zasadę odwracania zależności, podać przykład jej zastosowania.
73. Omówić pojęcia sprzężenia (coupling) i spistości (cohesion) podsystemów oraz ich wpływ na koszt utrzymania oprogramowania.

Inżynieria systemów

74. Sprawdzić, czy podana relacja przy określonych założeniach jest w 3PN, PNBC, 4PN; doprowadzić relację do wybranej postaci normalnej.
75. Wyjaśnić strukturę indeksów typu drzewo B+ oraz indeksów haszowych; podać przykład zastosowania indeksów w bazach danych.
76. Omówić podany poziom izolacji transakcji (read uncommitted, read committed, repeatable read, serializable), opisać jego przykładową implementację; podać wynik określonego harmonogramu, przy założeniu, że transakcje działają w wybranych poziomach izolacji.
77. Napisać zdanie SQL, które będzie wybierać dane zgodnie z zadanymi kryteriami; należy wykorzystać wybrane klauzule SELECT, FROM, WHERE, GROUP BY, HAVING, OVER oraz odpowiedni operator złączenia tabel lub podzapytanie.
78. Wyjaśnić pojęcie szeregowalności harmonogramów transakcji w bazach danych; podać przykład harmonogramu szeregowalnego oraz harmonogramu, który nie jest szeregowalny.
79. Omówić działanie i schemat budowy podanego cyfrowego układu kombinacyjnego.

80. Omówić działanie i schemat budowy podanego cyfrowego układu sekwencyjnego.
81. Przy pomocy wybranej metody zminimalizować podaną funkcję logiczną.
82. Przeanalizować zmiany stanów procesów w systemie operacyjnym wykorzystującym podany algorytm szeregowania z wywłaszczaniem.
83. Zaproponować rozwiązanie podanego problemu synchronizacji procesów w oparciu o mechanizmy muteksu, semafora, monitora.
84. Opisać działanie procesu (w szczególności dostęp do pamięci) w systemie operacyjnym wykorzystującym pamięć wirtualną.
85. Omówić organizację fizyczną i logiczną systemu plików oraz podstawowe komendy UNIX-owe dotyczące obsługi plików i katalogów (`cp`, `rm`, `mkdir`, `rmdir`, `cd`, `ln`).
86. Opisać model ISO OSI; opisać protokoły IP, TCP oraz UDP; objaśnić cel stosowania oraz znaczenie określonych (najważniejszych) pól w nagłówkach IPv4, IPv6, TCP, UDP.
87. Podać zasady adresacji klasowej i bezklasowej IPv4; dla podanego bezklasowego adresu IPv4 i podanej maski podać adres sieci, adresy hostów w tej sieci oraz adres rozgłoszenia; opisać zwięźle protokoły IPv6;
88. Opisać najważniejsze procesy zachodzące w sieci komputerowej od momentu wpisania adresu strony WWW do wyświetlenia strony w oknie przeglądarki (komunikat HTTP, segment TCP, system DNS, pakiet IP, ARP, ramka).
89. Opisać działanie przełączników Ethernet, opisać co to są sieci VLAN, oraz opisać cel stosowania protokołu STP; podać jak mogłoby dojść do burzy rozgłoszeń ramek - broadcast storm.
90. Opisać działanie routerów, objaśnić co to jest routing statyczny oraz zwięźle opisać i porównać protokoły routowania RIP oraz OSPF (opisać sposób rozgłaszania informacji o sieciach, objaśnić jak tworzone są wpisy w tablicach routowania).
91. Objaśnić, jakie metody szyfrowania są wykorzystywane w sieciach komputerowych i podać do czego są stosowane (w szczególności opisać metody z kluczem symetrycznym oraz z kluczem publicznym i prywatnym); opisać jak działają podpisy cyfrowe oraz co to są certyfikaty cyfrowe.
92. Wyjaśnić, czym są wirtualne sieci prywatne; zwięźle opisać protokoły IPSec (AH, ESP w trybie tunelowym i w trybie transportu).