

Перелік теоретичних екзаменаційних питань з дисципліни «Алгоритми і методи обчислень»

1. Неформальне тлумачення алгоритму.
2. Історія поняття алгоритму.
3. Визначення алгоритму.
4. Основні характеристики алгоритму.
5. Формалізація поняття алгоритму.
6. Введення в теорію алгоритмів.
7. Формалізація поняття алгоритму. Формальні алгоритмічні системи(ФАС).
8. Скінченний автомат. Детермінований скінченний автомат(DFA). Недетермінований скінченний автомат(NFA). Магазинний автомат(PDA).
9. Формальні граматики та формальні мови.
10. Методи відображення алгоритмів.
11. Вербальне подання алгоритму.
12. Подання алгоритму псевдокодом з використанням формальних мов.
13. Схематичне подання алгоритму.
14. Покрокове проектування алгоритмів.
15. Підходи при синтезі алгоритмів(алгоритмічні стратегії).
16. Повний перебір.
17. Бектрекінг (перебір з поверненням).
18. Метод “розділяй і пануй”.
19. Метод гілок і границь.
20. Скупі алгоритми.
21. Динамічне програмування.
22. Оцінка розміру вхідних даних. Складність по пам’яті. Складність по часу виконання алгоритму.
23. Порівняння найкращих, середніх та найгірших оцінок.
24. Ріст функцій. O -, o -, W -, w -, Q - нотації. Стандартні класи ефективності алгоритмів.
25. Математичний аналіз нерекурсивних алгоритмів.
26. Рекурсивні алгоритми. Математичний аналіз рекурсивних алгоритмів.
27. Емпіричний аналіз алгоритмів.
28. Визначення P - і NP - класи складності. NP -повнота (теорема Кука). Стандартні NP -повні проблеми.
29. Алгоритми пошуку.
30. Алгоритми сортування даних.
31. Алгоритми порівняння зі взірцем.
32. Чисельні алгоритми.
33. Графи та мережеві алгоритми.
34. Паралельні та розподілені алгоритми.
35. Застосування базових алгоритмів з використанням STL(Standart Template Library) при узагальненому програмуванні на C++.
36. Застосування базових алгоритмів з використанням JCL(Java Class Library) при узагальненому програмуванні на Java.

37. Застосування базових алгоритмів з використанням FCL(Framework Class Library) при програмуванні на C#.
38. Застосування алгоритмів для вирішення задач лінійної алгебри з використанням uBLAS при програмуванні на C++.
39. Застосування простих алгоритмів обробки сигналів з використанням OpenCV(Open Source Computer Vision Library) при програмуванні на C++.
40. Послідовні формальні алгоритмічні системи еквівалентні машині Тюрінга. Теза Черча.
41. Нормальні алгоритми Маркова.
42. Регістрова машина.
43. РАМ-машина.
44. Функційні моделі обчислень.
45. Лямбда числення.
46. Типізоване лямбда числення.
47. Рекурсивні функції.
48. Комбінація логіки.
49. Клітковий автомат.
50. Абстрактна перезаписуюча система.
51. Паралельні моделі обчислень.
52. ПРАМ-машина. ПРАМ-машина на основі ПВДН.
53. Мережа процесів Кана.
54. Мережа Петрі.
55. Синхронний потік даних та парадигма реактивного програмування.
56. Мережа взаємодій.
57. Парадигма функційного програмування.
58. Парадигма реактивна програмування.
59. Основні поняття і принципи квантових обчислень. Складність квантових обчислень.
60. Кубіти, квантові вентиля та квантові регістри.
61. Функціонування квантової системи.
62. Важливі алгоритми квантових обчислень.
63. Алгоритм Шора.
64. Алгоритм Гровера.

Можливі варіанти екзаменаційних завдань з дисципліни
«Алгоритми і методи обчислень»

Завдання 1-7

- 1 Нормальні алгоритми Маркова. Навести приклад реалізації видалення голосних літер у стрічці “abacaba” для алфавіту $V = \{a, b, c\}$.
- 2 Машина Тюрінга. Приклад реалізації кон’юнкції ($Z = X \bullet Y$) над тетрадами $X = 1101$, $Y = 0010$.
- 4 Написати програму для рівнодоступної адресної машини (РАМ) для знаходження значення виразу $S = a_1 + a_2 + 1$.
- 5 Означити натуральні числа за допомогою нотацій Бекуса-Наура.
- 6 Означити цілі числа за допомогою регулярного виразу.
- 7 Структура бібліотеки STL.
- 8 Алгоритми бібліотеки STL.
- 9 Застосування базових алгоритмів при узагальненому програмуванні на Java.
- 10 Застосування базових алгоритмів при узагальненому програмуванні на C#.
- 11 Інтерфейс бібліотек підпрограм BLAS та три рівні його функціональності.
- 12 Структура бібліотеки OpenCV.
- 13 Парадигма функційного програмування. Приклад коду програми.
- 14 Парадигма реактивного програмування. Приклад коду програми.

Завдання 8-10

- 15 Навести приклад програми з використанням бібліотеки STL для контейнеру ***std::vector*** для визначення мінімального та максимального значення всього масиву та його заданої частини.
- 16 Навести приклад програми з використанням бібліотеки STL для сортування всього ***std::vector*** та заданої його частини.
- 17 Для завдання №15 або №16 замінити використання послідовного контейнеру ***std::vector*** на асоціативний контейнер ***std::map***.
- 18 З використанням бібліотеки ***uBLAS*** написати програму для розв’язання системи лінійних алгебраїчних рівнянь:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

- 19 З використанням ***std::regex*** написати програму, яка шукає в тексті лексеми заданого формату.
- 20 Розробити програму, яка за допомогою ***boost::spirit*** реалізовує наступний синтаксис для запису виразу:

<pre><expression> = <term_a>, { ("+" "-"), <term_a> }*. <term_a> = <term_m>, { ("*" "/"), <term_m> }*. <term_m> = <value> "+" <term_m> "-" <term_m> "(", <expression>, ")".</pre>
--