

ПРОГРАММА АТТЕСТАЦИОННОГО ИСПЫТАНИЯ ПО ИНФОРМАТИКЕ (САУ)

I. НА 2 СЕМЕСТР.

ОБЩИЕ ВОПРОСЫ (БЕЗ ПРИВЯЗКИ К PYTHON)

1. Наивный поиск подстроки в строке.
2. Алгоритм обращения чисел в массиве.
3. Алгоритм циклического сдвига в массиве.
4. Поиск корня уравнения методом бисекции. Требования алгоритма к функции.
5. Поиск значения в упорядоченном массиве методом бисекции.
6. Сортировка обезьяны. Применимость и асимптотика алгоритма.
7. Сортировка вставками. Применимость и асимптотика алгоритма.
8. Сортировка выбором. Применимость и асимптотика алгоритма.
9. Сортировка методом пузырька. Применимость и асимптотика алгоритма.
10. Сортировка дурака. Применимость и асимптотика алгоритма.
11. Сортировка подсчётом. Применимость и асимптотика алгоритма.
12. Поразрядная сортировка. Применимость и асимптотика алгоритма.
13. Быстрая сортировка Хоара. Применимость и асимптотика алгоритма.
14. Сортировка слиянием. Применимость и асимптотика алгоритма.
15. Рекурсия. Прямой и обратный ход рекурсии. Стек вызовов при рекурсии.
16. Алгоритм Евклида. Реализация через цикл и через рекурсию.
17. Быстрое возведение в степень. Асимптотика алгоритма.
18. Вычисление чисел Фибоначчи. Реализация через цикл и через рекурсию.
19. Ханойские башни.
20. Динамическое программирование. Сходство с рекурсией и отличие от неё. Когда рекурсия применима, а динамическое программирование нет.
21. Двумерное динамическое программирование. Задача о количестве траекторий шахматного короля.
22. Наибольшая общая подпоследовательность. Асимптотика алгоритма.
23. Наибольшая возрастающая подпоследовательность. Асимптотика алгоритма.
24. Односвязные списки и массивы.
25. Стек. Использование стека для проверки корректности скобочной последовательности.
26. Двусвязный список. Очередь.
27. Пирамида (куча). Асимптотика добавления и удаления элемента в кучу.
28. Пирамидальная сортировка. Асимптотика алгоритма.
29. Открытая и закрытая хеш-таблица. Описать добавление элемента. Асимптотика поиска.

ВОПРОСЫ ПО СИНТАКСИСУ ЯЗЫКА PYTHON 3

1. Основные конструкции языка: бинарные и унарные операторы, циклы for и while, оператор ветвления.
2. Ссылочная модель. Проблема копирования массива. Способы решения.
3. Синтаксис описания функций. Именованные параметры. Проблема передачи пустого списка как значения по умолчанию. Способы решения.

4. Строки и операции над ними. Срезы.
5. Стандартные типы данных: list, set, dict. Операции и их асимптотика.
6. Кортежи переменных и множественное присваивание. Отличие кортежей от списков.
7. Использование модулей. Импорт. Пример использования модуля random.
8. Списки в Python. List comprehensions: генерация списков.

II. НА 3 СЕМЕСТР.

ОБЩИЕ ВОПРОСЫ

1. Определение графа. Степень вершины, петли, кратные рёбра, инцидентность, смежные вершины.
2. Вершины и рёбра. Пути и циклы в графах.
3. Связность графов. Компоненты связности. Сильная и слабая связность.
4. Определение дерева. Свойства дерева. Остовное дерево графа.
5. Взвешенный граф. Минимальное остовное дерево. Алгоритм Прима.
6. Способы представления графа в памяти: список рёбер, матрица смежности, списки смежности.
7. Поиск в глубину и поиск в ширину.
8. Алгоритм Косарайю поиска компонент сильной связности орграфа.
9. Выделение компонент связности несвязного графа.
10. Алгоритм Дейкстры. Асимптотика при различных реализациях.
11. Алгоритм Флойда-Уоршелла. Асимптотика алгоритма и преимущества перед алгоритмом Дейкстры.
12. Эйлеров цикл. Эйлеров путь. Эйлеровы графы.
13. Гамильтонов цикл. Гамильтоновы графы. Задача о коммивояжёре.
14. Топологическая сортировка. Алгоритм Тарьяна.
15. Корневые деревья и их хранение в памяти.
16. Двоичное дерево поиска. Хранение двоичного дерева поиска в памяти.
17. Асимптотика поиска, добавления и удаления элемента в двоичное дерево поиска.
18. Сбалансированное дерево поиска. Балансировка деревьев поиска. Левое и правое малое и большое вращение.
19. Вычисление расстояния Левенштейна.
20. Однозначное декодирование неравномерного кода. Условие Фано. Префиксный бор двоичного кода.
21. Z-функция строки. Наивное вычисление и его оптимизация.
22. Префикс-функция строки. Наивное вычисление и алгоритм Кнута-Морриса-Пратта
23. Машина Алана Тьюринга. Определение и примеры алгоритмов.

III. НА 4 СЕМЕСТР

1. История эволюции вычислительных систем, основные функции, выполняемые современными операционными системами, принципы их внутреннего построения;
2. Концепция процессов в операционных системах;
3. Основные алгоритмы планирования процессов;
4. Логические основы взаимодействия процессов;
5. Концепция нитей исполнения и их отличие от обычных процессов;
6. Программные алгоритмы организации взаимодействия процессов и предъявляемые к ним требования;
7. Основные механизмы синхронизации в операционных системах;
8. Основные принципы управления файловыми системами;

9. Организацию управления устройствами ввода-вывода на уровне как технического, так и программного обеспечения, основные функции подсистемы ввода-вывода;
10. Основные проблемы безопасности операционных систем и подходы к их решению.
11. Идеология объектно-ориентированного подхода;
12. Принципы программирования структур данных для современных программ; типовые решения, применяемые для создания программ;

IV. НА 5 СЕМЕСТР

1. Event-driven и message-driven программирование на примере XWindows Widgets, Mac OS X Interface Builder и подсистемы GDI MS Windows.

Event-driven и message-driven программирование на примере XWindows Widgets, Mac OS X Interface Builder и подсистемы GDI MS Windows. Принципы агрегирования COM-объектов. Принципы поддержки ООП в операционных системах и языках, общее и различное. Реализация исключений в языке C++ и в ОС Windows.

2. Адресное пространство приложения: куча, стек и статические объекты. Адресное пространство приложения: куча, стек и статические объекты. Динамическая инициализация и клонирование объектов. Хранение объектов в адресном пространстве. Виртуальные функции. Наследование абстрактных классов в C++, чисто виртуальные функции.

3. Базовые основы элементарной техники программирования. Базовые основы элементарной техники программирования. Технические основы программной реализации формальных структур данных, итераторы. Списки, очереди, стеки, наборы, упорядоченные наборы, массивы, деревья, хранение графов, hash-таблицы; примеры использования структур данных, как выбрать структуру, соответствующую задаче. Применение hash-таблиц.

4. Безопасность ПО.

Безопасность ПО. Написание программ без «дырок». Техника кодирования защищенных программ и типичные ошибки. Переполнение буфера, определение уровня доступа, работа с минимально возможными привилегиями, криптография и ее корректное применение, предохранение секретных данных, работа с входными данными, проблемы разных путей доступа к одним и тем же данным, запросов к базам данных и веб-страницам, а также проблемы поддержки интернационального ПО. Моделирование угроз. Классификация опасностей STRIDE – Spoofing (подмена данных), Tampering (подделка и изменение содержания данных), Repudiation (незаконный отказ от проведенной операции), Information disclosure (разглашение информации), Denial of service (отказ в обслуживании), Elevation of privilege (незаконное поднятие привилегий). Методика оценки риска DREAD – Damage potential (что может быть сломано), Reproducibility (повторяемость), Exploitability (пригодность угрозы для использования), Affected users (на каких пользователей повлияет), Discoverability (возможно ли детектировать факт использования).

5. Динамическая идентификация и приведение типов (RTTI). Динамическая идентификация и приведение типов (RTTI). Обработка исключительных ситуаций. Разные способы «универсализации» алгоритмов – от абстрактных типов данных «ADT» до стандартной библиотеки шаблонов (Standard Template Library) языка программирования C++ (STL).

6. Краткий обзор ООП реализации в языке C++. Краткий обзор ООП реализации в языках C++. Интерфейсы, полиморфизм и перегрузка операторов в C++. Параметризованные классы. Дружественные функции. Потоки ввода-вывода в C++. Наследование как один из вариантов комбинирования объектов.

7. Краткий сравнительный обзор ООП реализации в языках C++ и ObjectiveC, позднее и раннее связывание. Краткий сравнительный обзор ООП реализации в языках C++ и ObjectiveC, позднее и раннее связывание. Техническая организация ООП поддержки языков программирования для позднего связывания.

8. Параллельное программирование.

Параллельное программирование. Параллельные программы – от работы с разделяемой памятью, использования массивно-параллельных компьютеров и до распределенных расчетов на многих физических компьютерах. Декомпозиция задач на параллельные куски. Параллелизм данных, параллелизм кода. Паттерны параллельного программирования: параллелизм на уровне задач – декомпозиция задачи, «разделяй и властвуй» – декомпозиция задач и данных, геометрическая декомпозиция – декомпозиция данных, конвейерное исполнение – декомпозиция потока данных, «фронт волны» – декомпозиция данных с «многомерными» зависимостями. Пример типового шаблона программирования – пул нитей.

9. Принципы и философия ООП в языках, программных системах и операционных системах. Принципы и философия ООП в языках, программных системах и операционных системах. Инкапсуляция, полиморфизм и наследование/агрегирование. Динамический и статический подход в описании классов.

10. Проблемы, специфические для параллельного исполнения многонитевых программ. Проблемы, специфические для параллельного исполнения многонитевых программ – синхронизация (между несколькими нитями), коммуникация (проблемы полосы пропускания и задержек, связанных с обменом данными между нитями), балансировка нагрузки (между нитями), масштабируемость (эффективность использования многих нитей). Написание высокопроизводительных программ, оценка условий и выбор способов реализации.

11. Процесс написания программ. Процесс написания программ. Оформление текстов, требования к текстам и комментариям. Сопровождение программ. Документация на ПО, SDP (software development plan – план разработки ПО).

12. Работа с разделяемой памятью.

Работа с разделяемой памятью. Синхронизационные примитивы (низкоуровневые атомарные команды, критические секции, взаимоисключающая блокировка – mutex, рекурсивная блокировка – lock, блокировка чтения-записи – read/write lock, многопроцессорная блокировка – spinlock, семафоры, барьеры). Реализация одних примитивов через другие, относительная «мощность» примитивов. Задача о консенсусе как способ оценки примитивов.

13. Техническая специфика параллельных программ.

Техническая специфика параллельных программ – производительность, условия «гонки» – race condition, взаимная блокировка – deadlock, повторно-входимые программы, потоко-безопасные (thread-safe) библиотеки. Неблокирующие примитивы синхронизации, транзакционная память. Организация высокопроизводительных параллельных вычислений.

14. Эволюция современного аппаратного обеспечения и ее влияние на программное обеспечение. Эволюция современного аппаратного обеспечения и ее влияние на программное обеспечение. Гипертредовые (многопоточные) и многоядерные процессоры, универсальные графические (GPU) процессоры и новые возможности по их использованию

V. НА 6 СЕМЕСТР

1. Наивный поиск подстроки в строке.
2. Алгоритм обращения чисел в массиве.
3. Алгоритм циклического сдвига в массиве.
4. Поиск корня уравнения методом бисекции. Требования алгоритма к функции.

5. Поиск значения в упорядоченном массиве методом бисекции.
6. Сортировка обезьяны. Применимость и асимптотика алгоритма.
7. Сортировка вставками. Применимость и асимптотика алгоритма.
8. Сортировка выбором. Применимость и асимптотика алгоритма.
9. Сортировка методом пузырька. Применимость и асимптотика алгоритма.
10. Сортировка дурака. Применимость и асимптотика алгоритма.
11. Сортировка подсчётом. Применимость и асимптотика алгоритма.
12. Поразрядная сортировка. Применимость и асимптотика алгоритма.
13. Быстрая сортировка Хоара. Применимость и асимптотика алгоритма.
14. Сортировка слиянием. Применимость и асимптотика алгоритма.
15. Рекурсия. Прямой и обратный ход рекурсии. Стек вызовов при рекурсии.
16. Алгоритм Евклида. Реализация через цикл и через рекурсию.
17. Быстрое возведение в степень. Асимптотика алгоритма.
18. Вычисление чисел Фибоначчи. Реализация через цикл и через рекурсию.
19. Ханойские башни.
20. Динамическое программирование. Сходство с рекурсией и отличие от неё. Когда рекурсия применима, а динамическое программирование нет.
21. Двумерное динамическое программирование. Задача о количестве траекторий шахматного короля.
22. Наибольшая общая подпоследовательность. Асимптотика алгоритма.
23. Наибольшая возрастающая подпоследовательность. Асимптотика алгоритма.
24. Односвязные списки и массивы.
25. Стек. Использование стека для проверки корректности скобочной последовательности.
26. Двусвязный список. Очередь.
27. Пирамида (куча). Асимптотика добавления и удаления элемента в кучу.
28. Пирамидальная сортировка. Асимптотика алгоритма.
29. Открытая и закрытая хеш-таблица. Описать добавление элемента. Асимптотика поиска.

VI. НА 7 СЕМЕСТР

1. Определение графа. Степень вершины, петли, кратные рёбра, инцидентность, смежные вершины.
2. Вершины и рёбра. Пути и циклы в графах.
3. Связность графов. Компоненты связности. Сильная и слабая связность.
4. Определение дерева. Свойства дерева. Остовное дерево графа.
5. Взвешенный граф. Минимальное остовное дерево. Алгоритм Прима.
6. Способы представления графа в памяти: список рёбер, матрица смежности, списки смежности.
7. Поиск в глубину и поиск в ширину.
8. Алгоритм Косарайю поиска компонент сильной связности орграфа.
9. Выделение компонент связности несвязного графа.
10. Алгоритм Дейкстры. Асимптотика при различных реализациях.
11. Алгоритм Флойда-Уоршелла. Асимптотика алгоритма и преимущества перед алгоритмом Дейкстры.
12. Эйлеров цикл. Эйлеров путь. Эйлеровы графы.
13. Гамильтонов цикл. Гамильтоновы графы. Задача о коммивояжёре.
14. Топологическая сортировка. Алгоритм Тарьяна.
15. Корневые деревья и их хранение в памяти.
16. Двоичное дерево поиска. Хранение двоичного дерева поиска в памяти.
17. Асимптотика поиска, добавления и удаления элемента в двоичное дерево поиска.

18. Сбалансированное дерево поиска. Балансировка деревьев поиска. Левое и правое малое и большое вращение.
19. Вычисление расстояния Левенштейна.
20. Однозначное декодирование неравномерного кода. Условие Фано. Префиксный бор двоичного кода.
21. Z-функция строки. Наивное вычисление и его оптимизация.
22. Префикс-функция строки. Наивное вычисление и алгоритм Кнута-Морриса-Пратта
23. Машина Алана Тьюринга. Определение и примеры алгоритмов.

VII. НА 8 СЕМЕСТР

1. История эволюции вычислительных систем, основные функции, выполняемые современными операционными системами, принципы их внутреннего построения;
2. Концепция процессов в операционных системах;
3. Основные алгоритмы планирования процессов;
4. Логические основы взаимодействия процессов;
5. Концепция нитей исполнения и их отличие от обычных процессов;
6. Программные алгоритмы организации взаимодействия процессов и предъявляемые к ним требования;
7. Основные механизмы синхронизации в операционных системах;
8. Основные принципы управления файловыми системами;
9. Организацию управления устройствами ввода-вывода на уровне как технического, так и программного обеспечения, основные функции подсистемы ввода-вывода;
10. Основные проблемы безопасности операционных систем и подходы к их решению.
11. Идеология объектно-ориентированного подхода;
12. Принципы программирования структур данных для современных программ; типовые решения, применяемые для создания программ;