**Задание на 4 семестр**

**Практическая работа № 1 Динамическое программирование и оптимизационные задачи**

**Цель работы:** Изучить и реализовать классические задачи, решаемые с помощью динамического программирования, и провести анализ эффективности решений.

**Задания:**

1. **Задача о рюкзаке (0/1 рюкзак):**

* Реализуйте алгоритм динамического программирования для задачи о рюкзаке, где предметы можно либо взять целиком, либо оставить.
* Проведите тестирование на различных наборах данных и проанализируйте эффективность алгоритма.

1. **Задача наибольшей общей подпоследовательности (LCS):**

* Реализуйте алгоритм для нахождения наибольшей общей подпоследовательности двух строк.
* Тестируйте алгоритм на различных парах строк и оцените его сложность и время выполнения.

1. **Алгоритм построения множества Парето (M-множество и N-множество Парето):**

* Реализуйте алгоритм построения M-множества и N-множества Парето для заданного набора точек в многомерном пространстве.
* Проведите тестирование алгоритма на различных наборах данных и визуализируйте результаты для K измерений.
* Проанализируйте время выполнения алгоритма и предложите способы оптимизации.

1. **Задача о выполнимости булевой формулы (SAT):\*\***

* Реализуйте алгоритм проверки выполнимости булевой формулы (например, метод DPLL).
* Протестируйте алгоритм на различных булевых формулах, задаваемых в конъюнктивной нормальной форме (КНФ).
* Оцените эффективность алгоритма и проведите сравнительный анализ на простых и сложных формулах.

**Отчёт должен включать:**

* Описание и реализацию каждого алгоритма.
* Результаты тестирования на различных наборах данных.
* Графики и таблицы, демонстрирующие время выполнения и эффективность решений.
* Сравнительный анализ алгоритмов и выводы о применении динамического программирования в решении различных задач.

**Практическая работа № 2 Алгоритмы обработки графов.**

**Цель работы**: Изучить и реализовать основные алгоритмы работы с графами.

1. **Обход графа в ширину (BFS):**

* Реализуйте алгоритм обхода графа в ширину.
* Проверьте работу алгоритма на графе с 9–12 вершинами.
* Выведите порядок обхода вершин.

1. **Обход графа в глубину (DFS):**

* Реализуйте алгоритм обхода графа в глубину.
* Постройте и отобразите дерево DFS для заданного графа.

1. **Алгоритм Косарайю:\***

* Реализуйте алгоритм для поиска компонент сильной связности в ориентированном графе.
* Протестируйте его на графе с несколькими компонентами сильной связности.

1. **Алгоритмы построения минимального остова: \***

* Реализуйте и сравните алгоритмы Краскала и Прима.
* Протестируйте их на графах разного размера и сравните быстродействие.

1. **Реализация DFS-леса для несвязного графа.\*\***
2. **Алгоритм Беллмана-Форда для поиска кратчайшего пути (с отрицательными весами).\*\***

**Дополнения:**

1. Методические указания к алгоритмам:

* **BFS и DFS:** граф может быть представленный как ориентированный или неориентированный, и от этого зависят результаты.
* **Косарайю:** алгоритм состоит из двух проходов DFS и инверсии графа.
* **Краскал и Прим:** остов строится только для связных графов (тестировать на неориентированных графах).
* **DFS-лес:** используется для несвязных графов, где обход охватывает несколько компонент связности.

1. Формат представления графа: список смежности илиматрица смежности.
2. Тестирование и визуализация: для проверки работы алгоритмов желательно выводить:

* Матрицу смежности или список смежности графа.
* Порядок обхода вершин (для BFS и DFS).
* Дерево DFS и компоненты сильной связности (для Косарайю).

1. Сравнение сложности алгоритмов: определить ожидаемую временную сложность.
2. Визуализация (дополнительно, для заинтересованных студентов).

**Отчёт должен включать:**

* Описание каждого алгоритма.
* Реализацию и тестовые данные.
* Результаты тестирования (графики, таблицы, визуализация).
* Сравнительный анализ алгоритмов (по времени выполнения, сложности и другим критериям) и выводы

**Практическая работа № 3 Потоки в графах и вычислительная геометрия**

**Цель работы:** Изучить и реализовать алгоритмы работы с потоками в графах и алгоритмы вычислительной геометрии.

**Вариант 1: Потоки в графах**

1. **Алгоритм Форда-Фалкерсона:**

* Реализуйте алгоритм для поиска максимального потока в транспортной сети.
* Протестируйте алгоритм на графах с 6–8 вершинами.

1. **Алгоритм Беллмана-Форда:**

* Реализуйте алгоритм для поиска кратчайшего пути в графе с отрицательными весами.
* Проверьте работу алгоритма на графах с отрицательными циклами и без них.

1. **Дополнительно: Алгоритм Басакера-Гоуэна:\*\***

* Реализуйте алгоритм для нахождения минимального максимального потока в транспортной сети.
* Сравните его результаты с алгоритмом Форда-Фалкерсона.
* Протестируйте на небольших графах (6–8 вершин).

**Вариант 2 Вычислительная геометрия**

1. **Построение выпуклой оболочки:**

* Реализуйте алгоритмы Грэхема и Джарвиса.
* Протестируйте их на одинаковых наборах точек.
* Сравните эффективность алгоритмов

1. **Алгоритм Чана:**

* Реализуйте алгоритм для построения выпуклой оболочки 𝑂(𝑛 log𝑛 ).
* Сравните результаты работы с алгоритмами Грэхема и Джарвиса.

1. **Дополнительно: алгоритм Бентли-Оттмана:\***

* реализуйте алгоритм для поиска пересечений отрезков 𝑂(𝑛 log𝑛 ).
* Протестируйте алгоритм на случайных наборах отрезков.

**Отчет должен включать:**

* Краткое описание алгоритмов.
* Реализацию алгоритмов.
* Результаты тестирования на разных наборах данных.
* Графики и таблицы для демонстрации времени выполнения.