

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МУРМАНСКИЙ АРКТИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра информационных технологий

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ВЫПОЛНЕНИЮ ИТОГОВОГО ПРОЕКТА
по дисциплине
«Структуры и алгоритмы обработки данных»**

для направления подготовки 09.03.01 Информатика и вычислительная техника
направленность (профиль) «Информатика и вычислительная техника»

Мурманск
2023

Общие организационно-методические указания

Итоговый проект по дисциплине «Структуры и алгоритмы обработки данных» обучающиеся выполняют в пятом семестре. Целью выполнения проекта является изучение принципов разработки эффективных алгоритмов обработки данных и получение навыков анализа сложности используемых алгоритмов.

Результатом выполнения проекта являются работающее программное приложение, представленное в электронном виде и исходным кодом, пояснительная записка и презентация, представляющая итоги выполнения проекта.

Выполнение проекта направлено на формирование у обучающегося элементов общепрофессиональной компетенции: способен разрабатывать алгоритмы и программы, пригодные для практического применения (ОПК-8).

В ходе выполнения итогового проекта необходимо:

- выполнить обзор существующих алгоритмов решения поставленной в рамках курсового проектирования задачи;
- провести сравнительный анализ возможных алгоритмов решения задачи с указанием их сложности и ограничений в использовании;
- обосновать выбор конкретного алгоритма для решения поставленной задачи;
- программно реализовать алгоритм в рамках программного средства;
- разработать интуитивно понятный пользовательский интерфейс для программного средства с возможности графического отображения этапов или результатов работы программного средства;
- провести эксперименты по оценке практической сложности используемого алгоритма, сравнить с теоретическими оценками;
- сделать вывод об эффективности используемого алгоритма.

Примерный перечень тем

Вариант № 1. Сбалансированное дерево поиска (AVL-дерево) [4, 9, т.3, 13].

Разработать программу, демонстрирующую работу со сбалансированным деревом поиска (AVL-деревом).

Реализовать алгоритмы поиска с включением в AVL-дерево и удаления элементов из AVL-дерева.

Снабдить программу графической иллюстрацией. Отразить процесс балансировки в «замедленном режиме» с возможностью повтора.

Входная информация: текстовый файл, содержащий последовательность ключей. Также предусмотреть возможность ввода информации с экрана.

Вариант № 2. В-дерево, В+дерево [9, т.3, 13].

Разработать программу, демонстрирующую работу с В-деревом, В+деревом.

Реализовать алгоритмы поиска с включением в В-дерево и В+дерево и удаления элементов из В-дерева и В+дерева.

Снабдить программу графической иллюстрацией. Отразить процесс балансировки в «замедленном режиме» с возможностью повтора.

Входная информация: текстовый файл, содержащий последовательность ключей. Также предусмотреть возможность ввода информации с экрана.

Вариант № 3. Эффективность методов внешней сортировки [13, 20]

Разработать программу, демонстрирующую работу методов внешней сортировки. В зависимости от выбора пользователя программа реализует методы: поглощения, многопутевого челночного слияния.

Изменяя размер исходного файла (10, 100, 1000, 10000,... блоков), а также размер блока, для каждого метода составить таблицу, демонстрирующую изменение времени сортировки в зависимости от размера блока и размера файла (с этой целью использовать таймер).

Сравнить эффективность методов.

Вариант № 4. Алгоритм Прюфера (формула Кэли о числе различных деревьев, которые могут быть построены на n вершинах) [11, 13].

Программно реализовать алгоритм получения кода Прюфера, позволяющий получить по данному дереву с n вершинами соответствующий ему вектор с $n - 2$ компонентами, и наоборот по данному вектору с $n - 2$ компонентами соответствующее ему дерево с n вершинами. Снабдить программу графической иллюстрацией.

Входная информация: текстовый файл, содержащий в зависимости от решаемой подзадачи:

- либо информацию о дереве (вершина + ссылки на всех её сыновей),
- либо натуральные числа – компоненты вектора.

Вариант № 5. Поиск выхода из лабиринта

Разработать программу, позволяющую найти в графе маршрут, который начинается в заданной вершине v_0 и наверняка приводит в другую заданную вершину v_1 (выход).

Программно реализовать методы Винера, Терри и радиальный метод. Возможно, метод Ли

Снабдить программу меню, которое позволит пользователю выбрать метод поиска выхода из лабиринта.

Снабдить программу графической иллюстрацией, отражающей пошаговое движение по лабиринту.

Организовать формирование исходного графа, как случайным образом, так и путём считывания из текстового файла.

Вариант № 6. Нахождение максимального потока в сети [11, 13].

Разработать программу, реализующую алгоритм нахождения максимального потока в сети на основе алгоритма Голдберга–Тарьяна и Эдмондса–Карпа..

Снабдить программу графической иллюстрацией.

На входе: текстовый файл, содержащий сеть $F(V; E; \varphi)$ с источником $s \in V$ и стоком $f \in V$, матрицу пропускных способностей дуг сети.

На выходе: матрица максимального потока.

Вариант № 7. Нахождение максимального потока в сети [11, 13].

Разработать программу, реализующую алгоритм нахождения максимального потока в сети на основе алгоритма расширения предпотока (с различными параметрами).

Снабдить программу графической иллюстрацией.

На входе: текстовый файл, содержащий сеть $F(V; E; \varphi)$ с источником $s \in V$ и стоком $f \in V$, матрицу пропускных способностей дуг сети.

На выходе: матрица максимального потока.

Вариант № 8.

Реализация разреженных матриц и работа с ними [21, 22].

Снабдить программу пошаговой иллюстрацией.

На входе: текстовый файл, содержащий разреженные матрицы.

На выходе: визуализация произведения и результат.

Вариант № 9.

Оптимизация методов динамического программирования [23, 24].

Разработать программу, реализующую алгоритм четырёх русских для умножения матриц и для задачи о наибольшей общей подпоследовательности.

Снабдить программу графической иллюстрацией.

На входе: текстовые файлы с матрицами, либо строки, введённые с клавиатуры. Длины битовой маски также вводятся с клавиатуры.

Вариант № 10.

Методы динамического программирования [13, 14].

Разработать программу, реализующую алгоритмы решения задачи о редакционном расстоянии (расстоянии Дамерау-Левенштейна). Одним из вариантов реализации должен быть алгоритм Вагнера-Фишера.

Вариант № 11.

Алгоритм сортировки целых чисел (сортировка Хана – Y. Han) [15].

Разработать программу, реализующую алгоритм сортировки Хана.

Вариант № 12.

Реализация толстой кучи [16].

Разработать программу, реализующую алгоритмы работы с толстой кучей на избыточном счётчике.

Снабдить программу графической иллюстрацией.

Вариант № 13.

Реализация тонкой кучи [17, 18].

Разработать программу, реализующую алгоритмы работы с фибоначчией и тонкой кучей.

Снабдить программу графической иллюстрацией.

Вариант № 14.

Разработать программу, демонстрирующую работу с КЧ-деревом, Splay-деревом [9, т.3, 19].

Реализовать алгоритмы поиска с включением в КЧ-дерево и Сплей-дерево и удаления элементов из них.

Снабдить программу графической иллюстрацией. Отразить процесс балансировки в «замедленном режиме» с возможностью повтора.

Входная информация: текстовый файл, содержащий последовательность ключей. Также предусмотреть возможность ввода информации с экрана.

Основные требования к алгоритмической и программной реализации

Следует придерживаться следующих требований к алгоритмической и программной реализации:

- использовать технологию нисходящего структурного проектирования и программирования;
- построить для каждой задачи алгоритм в виде блок-схемы [1] или диаграммы Насси-Шнайдермана [8];
- если алгоритм можно разбить на последовательность логически законченных действий, то законченное действие оформить в виде подпроцесса;
- отделять функции по вводу-выводу от вычислительных операций;
- во всех задачах выводить на экран условия, исходные и результатные данные;
- разработать для каждой задачи набор тестов;
- программа должна сообщать обо всех ситуациях, когда она не в состоянии правильно обработать данные (обработка исключений).

Требования к программному комплексу:

- выполнить системы помощи для пользователя, предусмотреть окно, содержащее сведения о разработчике программного комплекса
- вывод на экран текста условия задачи;
- вывод на экран исходных и результатных данных;
- во всех заданиях реализовать визуализацию алгоритма;
- работоспособность комплекса доказать на контрольных примерах.

Функциональная структура приложения определяется на основе задач, которые необходимо решить в предложенном варианте задания. Функции объединяются в группы. Группы функций могут быть связаны между собой по признаку взаимного использования.

В пояснительной записке необходимо указать перечень функций разрабатываемого программного комплекса, например (примерный перечень):

- вывод на экран системы помощи;
- чтение файла с исходными данными;
- генерирование данных;
- вывод элементов массива на экран;
- поиск элемента по заданному критерию отбора;
- сортировка элементов массива;
- и т. п.

ПРИМЕР СТРУКТУРНОЙ ДИАГРАММЫ ПРОГРАММНОГО ПРИЛОЖЕНИЯ

В качестве примера на рис. 1 и 2 приведены примерные фрагменты структурной диаграммы программного приложения.

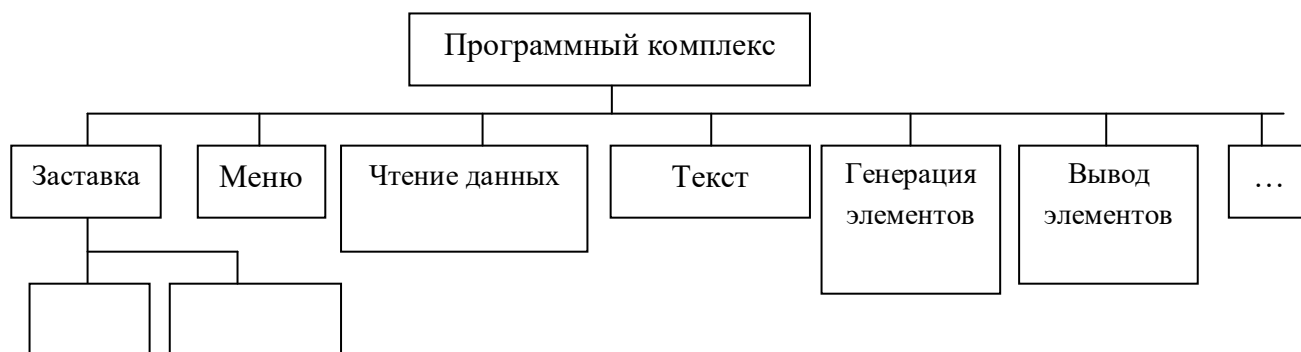


Рис. 1. Фрагмент структурной диаграммы программы, вариант 1

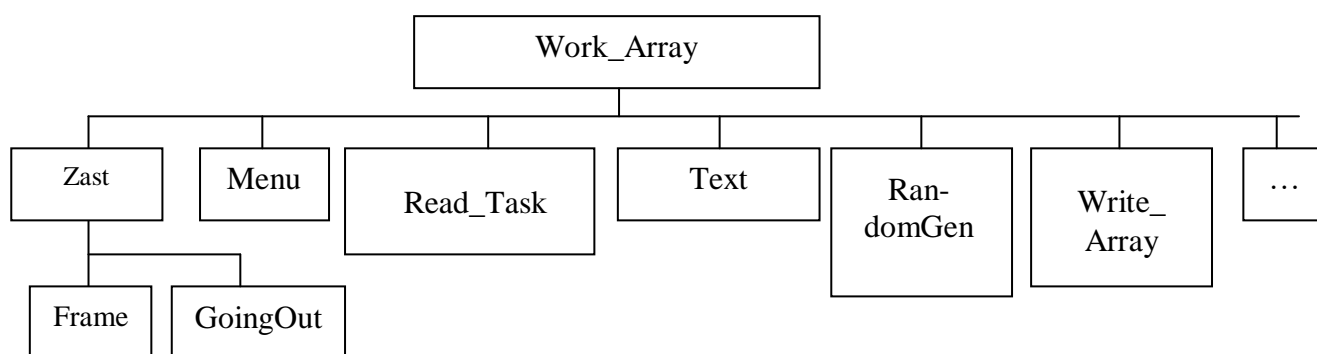


Рис. 2. Фрагмент структурной диаграммы программы, вариант 2

Между рис. 1 и 2 введены следующие соответствия:

Программный комплекс – главный модуль Work_Array;

Заставка – метод Text_Zas, выводит на экран в авторскую заставку;

Ожидание – метод GoingOut, ожидает нажатие любой клавиши;

Меню – метод Menu реализует выбор пользователем задачи;

Текст – метод Text выводит текстовые сообщения;

Генерация элементов – метод RandomGen используется для генерации входных данных.

Для одинаковых по функциональности подпрограмм блок-схема рисуется один раз. В программном комплексе подпрограммы оформляются как внешняя библиотека и используются во всех задачах.

Основные требования к содержанию пояснительной записки

Пояснительная записка должна включать в себя титульный лист, оглавление, введение, три главы, заключение, список источников и приложения.

Титульный лист оформляется в соответствии с Приложением 1.

Введение содержит цель проекта, краткое описание решаемой задачи и обоснование её практической значимости (указать из различных научных и прикладных сфер 4–5 примеров, при решении которых возникает необходимость решения данной задачи).

Первая глава содержит четыре параграфа: 1) четкую и ясную постановку задачи на естественном языке; 2) обзор приведенных в литературе, включая современные источники, алгоритмов решения задачи; 3) сравнительный анализ алгоритмов, рассматриваемых в предыдущем параграфе, с теоретическими оценками временной и пространственной сложности алгоритмов, возможных ограничений на входные данные; 4) обоснование выбора конкретного алгоритма (алгоритмов) для решения поставленной задачи и приводится его (их) блок-схема.

Вторая глава включает три параграфа и содержит описание процесса разработки программного средства, в рамках которого программно реализуется выбранный алгоритм. В первом параграфе исследуется и анализируется предметная область, приводится список требований к разрабатываемому программному средству; во втором параграфе описывается процесс проектирования программного средства, включая проектирование функций, проектирование данных, проектирование пользовательского интерфейса. В третьем параграфе описывается этап кодирования, при этом необходимо отразить физическую реализацию используемых структур данных, привести диаграмму классов и описать программной реализации основных функций и процедур. Особое внимание следует уделить графическому отображению результатов работы программы и (или) результатов конкретных этапов при решении задачи. В четвертом параграфе описывается этап тестирования разработанного программного средства с указанием методики тестирования, наборов тестовых данных, делается вывод о правильности и корректности работы программного средства на различных входных данных.

Третья глава посвящена экспериментальным оценкам сложности используемого на практике алгоритма обработки данных. В этой главе описывается постановка эксперимента и его результаты, приводится сопоставление результатов практической оценки сложности с теоретическими оценками.

В заключении делается вывод об эффективности используемых алгоритмов для решения поставленной задачи.

Список источников может включать в себя ссылки на учебные пособия, учебники, научные статьи. Ссылки на учебную литературу включают указание авторов, названия, издательства, год издания, количество страниц. Ссылки на статьи включают указание авторов, названия статьи, названия журнала, номер

выпуска, страницы, год выпуска. Такие источники, как справочные системы и руководства пользователя, указываются в подстраничных сносках.

Общие требования к оформлению текста пояснительной записки

В данном пункте описаны только основные требования к оформлению документа. Более подробно требования изложены Методических указаниях для обучающихся по направлениям подготовки укрупненной группы Информатика и вычислительная техника (уровень бакалавриата) «Правила оформления текстовых документов».

Основные требования следующие:

Гарнитура шрифта: Times New Roman Cyr.

Кегль шрифта: 14 – в основном тексте, в заголовках и подзаголовках в соответствии с уровнем.

Начертание шрифта заголовка и подзаголовков – полужирное, выделения в тексте – курсивным начертанием.

Заголовок и подзаголовки выравниваются по левому краю (заголовки первого уровня допустимо выравнивать по центру). В заголовке и подзаголовке запятые и точки не ставятся, если они выделены в отдельную строку (строки).

Нумерация страниц – вверху, справа (кроме титульного листа).

Поля страниц: слева, справа, сверху и снизу – по 2 см (при подшивке в папку рекомендуется левое поле – 3 см, правое – 1 см).

Абзацный отступ – 1,25, выравнивание абзаца – по ширине.

Межстрочный интервал в тексте – полуторный, в тексте с формулами допускается изменять интервал для удобства чтения формул.

Автоматическая расстановка переносов, максимальное число переносов подряд – 3.

Таблицы вставляются в текст с отступом после текста в 1 строку, выравниваются по центру и нумеруются. Перед таблицей помещается строка (выравнивание вправо), содержащая «Таблица N», далее на следующей строке следует название таблицы с большой буквы. При переносе таблицы на другую страницу необходимо вставить заголовок «Продолжение табл. N».

Рисунки вставляются в текст с отступом после текста в 1 строку, выравниваются по центру и нумеруются. Под рисунком помещается строка, содержащая «Рис. N.», далее в этой же строчке следует название рисунка с прописной буквы.

Критерии и шкала оценивания

Оценка	Критерии оценки
Отлично	Содержание проекта полностью соответствует заданию. Программ-

	<p>но реализованный алгоритм решения задачи работает корректно на произвольных (с учётом ограничений) входных данных. Пользовательский интерфейс продуман и интуитивно понятен. Программное средство графически без ошибок отображает результаты работы алгоритма. Программа демонстрирует устойчивую работу на тестовых наборах исходных данных, подготовленных обучающимся, обрабатывает все исключительные ситуации.</p> <p>В пояснительной записке представлены результаты обзора литературных и иных источников. Структура записки логически выдержана. Оформление записки отвечает требованиям, изложенным в методических указаниях.</p> <p>Проведён широкий ряд экспериментов по оценке эффективности алгоритма. По результатам экспериментов аргументированно обоснованы выводы.</p> <p>При защите проекта обучающийся правильно и уверенно отвечает на вопросы преподавателя, демонстрирует глубокое знание теоретического материала, способен аргументировать собственные утверждения и выводы</p>
Хорошо	<p>Содержание проекта полностью соответствует заданию. Программно реализованный алгоритм решения задачи работает корректно на входных данных, предоставленных обучающимся, но обрабатывает не все исключительные ситуации. Пользовательский интерфейс понятен, но требует улучшения. Графическое отображение программным средством результатов работы алгоритма имеет ряд недочётов, не влияющих на общее понимание работы алгоритма.</p> <p>В пояснительной записке представлены результаты обзора литературных и иных источников. Структура записки логически выдержана. Оформление записки отвечает требованиям, изложенным в методических указаниях.</p> <p>Проведено несколько экспериментов по оценке эффективности алгоритма. Выводы по результатам экспериментов недостаточно аргументированы.</p> <p>При защите проекта обучающийся правильно и уверенно отвечает на большинство вопросов преподавателя, демонстрирует хорошее знание теоретического материала, но не всегда способен аргументировать собственные утверждения и выводы. При наводящих вопросах преподавателя исправляет ошибки в ответе</p>
Удовлетворительно	<p>Содержание проекта не в полной мере соответствует заданию. Программно реализованный алгоритм решения задачи работает корректно на малом количестве частных случаев, не обрабатывает большинство исключительных ситуаций. Пользовательский интерфейс не продуман и понятен только разработчику. Графическое отображение программным средством результатов работы алгоритма имеет ряд существенных замечаний.</p> <p>Работа выполнена на основании 1–2 источников. Есть нарушения в логике изложения материала. Структура и оформление записки в целом соответствует требованиям, изложенным в методических указаниях.</p> <p>Не продумана схема эксперимента по оценке эффективности алгоритма. Аргументация выводов слабая или отсутствует.</p> <p>При защите проекта обучающийся допускает грубые ошибки при ответах на вопросы преподавателя, демонстрирует слабое знание</p>

	теоретического материала, в большинстве случаев не способен уверенно аргументировать собственные утверждения и выводы
Неудовлетворительно	<p>Содержание проекта в целом не соответствует заданию. Программное средство не разработано и/или находится в нерабочем состоянии (работает некорректно на любом наборе входных данных).</p> <p>Работа выполнена на основании 1 источника. Есть нарушения в логике изложения материала. Структура и оформление записки в целом не соответствует требованиям, изложенным в методических указаниях.</p> <p>Эксперименты по оценке эффективности не могут быть проведены из-за отсутствия работающего программного средства. Выводы отсутствуют.</p> <p>Защита проекта невозможна из-за отсутствия работающего программного средства.</p>

1. ГОСТ 19.701-90 (ИСО 5807-85). Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения [Электронный ресурс]. М. : Стандартинформ, 2010. 23 с. URL: <http://gostexpert.ru/gost/gost-19.701-90/download> (дата обращения: 02.12.2022).
2. ГОСТ Р 7.0.5-2008. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая ссылка. Общие требования и правила составления [Электронный ресурс]. М. : Стандартинформ, 2008. URL: <http://gostexpert.ru/gost/gost-7.0.5-2008> (дата обращения: 20.10.2022)
3. Окулов, С. М. Программирование в алгоритмах / С. М. Окулов - Москва : Лаборатория знаний, 2017. – 386 с. – Текст : электронный // ЭБС "Консультант студента" : [сайт]. – URL : <https://www.studentlibrary.ru/book/ISBN9785001014492.html> (дата обращения: 11.11.2022).
4. Вирт Н. Алгоритмы и структуры данных. М.: ДМК Пресс, 2010. 272 с.
5. Гагарина Л. Г., Колдаев В. Д. Алгоритмы и структуры данных [Электронный ресурс]. М.: Финансы и статистика, 2009. 304 с. URL: http://biblioclub.ru/index.php?page=book_red&id=225965&sr=1 (дата обращения: 20.09.2022).
6. Гроздева С. И., Терзиева Т. Ж. Статичные и динамичные средства Для визуализации методов сортировки массивов // Педагогическая информатика 1/2012. [Электронный ресурс]. URL:http://pedinf.ru/ARHIV/2012_1.pdf (дата обращения: 01.12.2022).
7. Златопольский, Д. М. Программирование : типовые задачи, алгоритмы, методы / Златопольский Д. М. - 4-е изд. - Москва : Лаборатория знаний, 2020. - 226 с. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <https://www.studentlibrary.ru/book/ISBN9785001017899.html> (дата обращения: 11.10.2022).
8. Макарова Н. В., Волков В. Б. Информатика: учеб. для вузов. СПб. : Питер, 2012. 576 с.
9. Кнут Д. Э. Искусство программирования. Том 1. Основные алгоритмы. Том 2. Получисленные алгоритмы. Том 3. Сортировка и поиск. Том 4А. : пер. с англ. М. : Изд-во «Вильямс», 2003, 2003, 2007, 2013.
10. Pisinger D. Algorithms for Knapsack Problems. – Datalogisk Institut København: DIKU-Rapport, Department of Computer Science, 1995.
11. S. Dasgupta, C. Papadimitriou, U. Vazirani Algorithms. – McGraw-Hill, 2006.

12. Дж. А. Андерсон Дискретная математика и комбинаторика. – М. : Издательский дом «Вильямс», 2004.
13. Т. Х. Кормен и др. Алгоритмы: построение и анализ. – М. : ООО «И.Д. «Вильямс»», 2013.
14. Романовский И.В. Дискретный анализ. СПб. : BHV, 2016.
15. Yijie Han Deterministic sorting in $O(n \log \log n)$ time and linear space // Journal of Algorithms, V. 50, Issue 1, 2004, Pages 96-105, ISSN 0196-6774.
16. Алексеев В., Таланов В. Структуры данных и модели вычислений. Лекция 9. Толстые кучи. // Интернет университет информационных технологий ИНТУИТ URL: <https://intuit.ru/studies/courses/100/100/lecture/2935?page=1> (дата обращения: 11.10.2022).
17. Алексеев В., Таланов В. Структуры данных и модели вычислений. Лекция 8. Тонкие кучи. // Интернет университет информационных технологий ИНТУИТ URL: <https://intuit.ru/studies/courses/100/100/lecture/2933> (дата обращения: 11.10.2022).
18. Гулаков В.К., Гулаков К.В. Дрожащая пирамида International Journal of Open Information Technologies ISSN: 2307-8162 vol. 6, no.1, 2018 URL : <http://www.injoit.org/index.php/j1/article/download/512/501> (дата обращения: 11.10.2022).
19. D. D. Sleator, R. E. Tarjan Self-Adjusting Binary Search Trees Journal of the Association for Computing Machinery. Vol. 32, No. 3, July 1985, pp. 652-686. URL : <http://www.cs.cmu.edu/~sleator/papers/self-adjusting.pdf> (дата обращения: 12.10.2022).
20. Мясникова Н.А. Алгоритмы и структуры данных. – М. : КНОРУС, 2021.
21. Писсанецки С. Технология разреженных матриц.: Пер. с англ. – М.: Мир, 1988.
22. Тьюарсон Р. Разреженные матрицы.: Пер. с англ. – М.: Мир, 1977.
23. Gregory V. Bard Accelerating Cryptanalysis with the Method of Four Russians // Cryptology ePrint Archive, Paper 2006/251 URL : <https://eprint.iacr.org/2006/251> (дата обращения: 05.11.2022).
24. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. – М. : Мир. 1979.

ПРИЛОЖЕНИЯ

П1. Пример титульного листа пояснительной записки

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МУРМАНСКИЙ АРКТИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра информационных технологий

Направление подготовки *09.03.01 Информатика и вычислительная техника*

ИТОГОВЫЙ ПРОЕКТ

по дисциплине «Структуры и алгоритмы обработки данных»

тема проекта

Студент: XXXXXXXXX XXXXXX XXXXXX, гр.

ФИО студента

Подпись

Дата сдачи

Руководитель:

Должность, ФИО

Подпись

Дата

Мурманск, 2023