**Требования к отчету о выполнении долгосрочного домашнего задания по учебной дисциплине «Технологии и методы программирования»**

Отчет представляет собой материал, который в обязательном порядке представляется обучающимся преподавателю в течение семестра. Он содержит описание результатов выполнения задания, обоснование принятых решений, а также доказательство работоспособности разработанной программы. Объем отчета обычно не превышает 10-20 страниц.

Принятый отчет о выполнении долгосрочного домашнего задания является критерием допуска к сдаче экзамена по учебной дисциплине.

**Структура отчета:**

Отчет оформляется в соответствии с методическими рекомендациями по оформлению письменных работ на кафедре, которые изложены в пособии ***«В.В. Абраменков, Л.В. Митрякова, В.В. Райх, В.А. Тихонов. Оформление письменных работ слушателей: учебно-методическое пособие. М.: Типография в/ч 33965, 2018, 132 с.»***

Отчет должен содержать:

1. Титульный лист;
2. Описание задания – в этом подразделе проводится полное описание задания, описывается суть задачи, анализ входных параметров, область их допустимых значений, анализ условий при которых задача имеет решение (не имеет решения), анализ ожидаемых результатов;
3. Основная часть:
   * математическая модель реализуемого алгоритма, включая его теоретическое описание и оценку его вычислительной сложности;
   * реализация алгоритма, включая листинг программы на языке программирования С++;
   * практическая оценка вычислительной сложности алгоритма (см. пояснения далее).
4. Выводы по результатам выполнения долгосрочного задания.

Для практической оценки вычислительной сложности алгоритма производится эксперимент по оценке времени выполнения алгоритма в зависимости от размера входных данных (количество узлов дерева, размер массива строк, количество вершин графа и т.п. в зависимости от варианта задания). Эксперимент проводится в несколько шагов – для различного размера входных данных, что позволяет практически убедиться в том, что время выполнения алгоритма соответствующим образом зависит от размера входных данных (линейная, квадратичная зависимость и т.д.). Размер данных подбирается индивидуально в зависимости от реализуемого алгоритма таким образом, чтобы измеренная скорость работы алгоритма составляла десятки секунд.

Количество шагов эксперимента выбирается самостоятельно, чтобы наглядно продемонстрировать вычислительную сложность алгоритма (Недостаточно использовать две точки для линейной сложности, через две точки можно провести кривую любой формы).

Измерения времени работы алгоритма необходимо производить любым известным программным способом, фиксируя в программе время начала и конца работы алгоритма, с последующим определением времени или с использованием счетчиков и т.п.

Для снижения погрешности экспериментов рекомендуется:

* Использовать версию RELEASE, а не DEBUG;
* каждый шаг эксперимента повторять несколько раз (не менее 5);
* измерения производить на одном и том же оборудовании, в одних и тех же условиях (режим работы, запущенное программное обеспечение). Рекомендуется отключить режимы энергосбережения, отключение неиспользуемых ядер процессора и т.п.

Результаты эксперимента необходимо представить в табличном и графическом виде. Графический вид позволяет наглядно отобразить результат. На графике необходимо совместить теоретическую и практическую оценку вычислительной сложности. Теоретическая изображается в виде кривой, практическая – в виде точек. Пример табличного представления приведен в таблице 1. Пример графического отображения – на рисунке 1.

Таблица 1 – Пример табличного представления результатов эксперимента.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Шаг** | **Кол-во элементов** | **Эксперимент №1, мс** | **Эксперимент №2, мс** | **Эксперимент №3, мс** | **Эксперимент №4, мс** | **Эксперимент №5, мс** |
| 1 | 100 | 160345 | 150028 | 170654 | 140253 | 180254 |
| 2 | 200 | 320054 | 330558 | 290658 | 340587 | 280564 |
| 3 | 300 | 500125 | 520145 | 600258 | 550658 | 580547 |
| 4 | 400 | 880547 | 900568 | 920587 | 911111 | 930258 |
| 5 | 500 | 1305847 | 1405874 | 1425588 | 1350258 | 1385473 |

Для удобства формирования графиков рекомендуется использовать табличный редактор MS Excel. Для построения теоретического графика необходимо аппроксимировать практические значения.

Рисунок 1 – Пример квадратичной зависимости, приведенной в таблице 1

Не должно быть сильных расхождений между практическими и теоретическими значениями, расхождение говорит либо о неправильно проведенном эксперименте, ошибках в алгоритме или неправильном построении теоретического графика. Пример неправильного графика приведен на рисунке 2.

Рисунок 2 – Пример неправильно построенного графика

**Варианты долгосрочных заданий**

| **№** | **Задание** | **Фамилия И.О** |
| --- | --- | --- |
|  | Бинарное дерево. Операция поиска |  |
|  | АВЛ-дерево. Операция поиска |  |
|  | Красно-черное дерево. Операция поиска |  |
|  | В-дерево. Операция поиска |  |
|  | Оптимальное бинарное дерево поиска. Операция поиска |  |
|  | Куча. Операция поиска |  |
|  | Хеш-таблица с разрешением коллизий с помощью связных списков. Операция поиска |  |
|  | Префиксное дерево. Операция поиска |  |
|  | Поиск подстроки в строке. Алгоритм Кнута-Морриса-Пратта |  |
|  | Поиск подстроки в строке. Алгоритм Бойера-Мура |  |
|  | Поиск подстроки в строке. Алгоритм Рабина-Карпа |  |
|  | Поиск подстроки в строке. Конечный автомат |  |
|  | Алгоритм сортировки «обменная» |  |
|  | Алгоритм сортировки «вставкой» |  |
|  | Алгоритм сортировки Шелла |  |
|  | Алгоритм сортировки «выбором» |  |
|  | Алгоритм пирамидальной сортировки |  |
|  | Алгоритм сортировки «слиянием» |  |
|  | Алгоритм быстрой сортировки |  |
|  | Поиск в ширину в неориентированном графе с помощью матрицы смежности |  |
|  | Поиск в глубину в неориентированном графе  с помощью матрицы смежности |  |
|  | Поиск в ширину в ориентированном графе с помощью списка смежности |  |
|  | Поиск в глубину в ориентированном графе  с помощью списка смежности |  |
|  | Построение минимального остовного дерева. Алгоритм Крускала |  |
|  | Построение минимального остовного дерева. Алгоритм Прима |  |
|  | Нахождение кратчайших путей в графе. Алгоритм А\* |  |
|  | Нахождение кратчайших путей в графе. Алгоритм Дейкстры |  |
|  | Выделение сильносвязных компонент в ориентированном графе |  |