

Министерство науки и образования РФ
Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
«Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

**Пояснительная записка к курсовой работе по теме
«Измерение временной сложности алгоритма
в эксперименте на ЭВМ»
по дисциплине
«Алгоритмы и структуры данных»
Вариант 40**

Выполнил студент гр.9308:

Аюпов Р.Н

Проверил:

Колинъко П.Г.

Оглавление

Введение.....	3
1. Задание.....	3
2. Формализация задания.....	3
3. Теоретические оценки временной сложности.....	4
4. Обработка результатов эксперимента.....	5
Вывод.....	7
Список используемых источников.....	8
Приложение.....	9

Введение

Исследование временной сложности алгоритмов над множествами последовательностями.

1. Задание

На основе программы, составленной по теме 3, можно выполнить статистический эксперимент по измерению фактической временной сложности алгоритма обработки данных.

2. Формализация задания

Программа из задания 3 дорабатывается таким образом, чтобы она генерировала множества мощностью, меняющейся, измеряла время выполнения цепочки операций над множествами и последовательностями и выводила результат в текстовый файл in.txt. Каждая строка этого файла должна по пару значений «размер входа — время» для каждого опыта. Затем эти данные обрабатываются, и по результатам обработки делается заключение о временной сложности алгоритма.

3. Теоретические оценки временной сложности

Вставка (insert)

В контейнере *multiset* элемент вставляется и в дерево и в последовательность.

Вставка в дерево без указания места начала поиска имеет логарифмическое время $O(\log n)$. Возможна вставка за константное время $O(1)$ если итератор вставки укажет для начала поиска места вставки на ключ, вставленный последним. Такая вставка допустима только для двуместных операций с множествами по схеме слияния.

Вставка в последовательность осуществляется за константное время $O(1)$ (при условии, что контейнер не расширяется при добавлении в него элемента, если происходит перераспределение, то само перераспределение является линейным по всему размеру).

Поэтому в среднем временная сложность алгоритма вставки для операций над множествами по схемам слияния $O(1)$, а для произвольной вставки сложность составляет $O(\log n)$.

Пересечение множеств (operator $\&=$)

Для объединения по схеме слияния происходит не более $2 \cdot (N_1 + N_2) - 1$ сравнений, где N_1 и N_2 — размеры контейнеров, т.е. временная сложность алгоритма $O(n)$.

Объединение множеств (operator $=$)

Временная сложность операции $O(n)$.

Разность множеств (operator $-$)

Временная сложность операции $O(n)$.

Симметрическая разность множеств (operator \wedge)

Временная сложность операции $O(n)$.

Укорачивание (erase)

Временная сложность операции $O(n \log n)$, т.к. операция формирует результат одновременно со вставкой в дерево, причем вставка идет с поиском от корня, т.е. за логарифмическое время.

Слияние (merge)

Временная сложность операции $O(n \log n)$.

Сцепление (concat).

Временная сложность операции $O(n \log n)$.

Таким образом, временная сложность цепочки операций $O(n \log n)$.

4. Обработка результатов эксперимента

Для эксперимента было выполнено 200 итераций. Мощность множеств генерировалась от 1 до 2000. Данные для эксперимента хранятся в файле in.txt

Результат работы программы RG41cb представлен на рисунке ниже.

Также он хранится в файле out.txt

201 data pairs received												
Dispersion	RMS	k	C	log N	N	N log N	N^2	N^3	N^4	Steps	Code	
4.62e-005	0.0068	1	0.0124	0	0	0	0	0	0	201	++	
1.02e-005	0.00319	2	-0.0375	0.00741	0	0	0	0	0	103	++	
8.49e-007	0.000921	2	-0.000124	0	1.19e-005	0	0	0	0	141	--	
8.48e-007	0.000921	3	0.000968	-0.000203	1.21e-005	0	0	0	0	292	--	
8.47e-007	0.00092	3	0.000215	0	8.42e-006	4.37e-007	0	0	0	318	--	
8.51e-007	0.000923	4	0.000294	-1.96e-005	8.71e-006	4.03e-007	0	0	0	637	--	
0.000323	0.018	3	-0.0614	0	0.000127	0	-4.56e-008	0	0	148	**	
7.34e-005	0.00857	4	-0.0644	0	0.00114	-0.000166	8.42e-008	0	0	319	--	
5.89e-006	0.00243	5	-0.1	0.03	-0.000921	0.000126	-4.36e-008	0	0	511	**	
0.000177	0.0133	4	-0.0635	0	0.000222	0	-1.92e-007	5.19e-011	0	310	--	
1.1e-005	0.00332	5	-0.0909	0.0216	-0.000107	0	7.55e-008	-1.74e-011	0	578	**	
4.18e-005	0.00647	5	-0.0664	0	0.00192	-0.000312	3.69e-007	-6.12e-011	0	512	**	
3.49e-006	0.00187	6	-0.107	0.0372	-0.00177	0.000261	-2.01e-007	2.7e-011	0	1085	--	
0.000112	0.0106	5	-0.065	0	0.000342	0	-5.11e-007	3.07e-010	-6.29e-014	478	**	
2.45e-005	0.00495	6	-0.0682	0	0.00297	-0.000528	1.1e-006	-4.12e-010	6.73e-014	981	--	
2.41e-006	0.00155	7	-0.114	0.0446	-0.00293	0.000461	-5.98e-007	1.81e-010	-2.6e-014	1720	**	

Уже по данному результату, мы можем судить, что временная сложность получилась линейной, т.к на это указывает последний столбец таблицы.

Результаты расчёта отношений выборочных дисперсий

Вспомогательная таблица. Определение подходящего уравнения регрессии															
Отношение дисперсий															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1,00	0,22	0,02	0,02	0,02	0,02	6,98	1,59	0,13	3,82	0,24	0,91	0,08	2,41	0,53
2	4,54	1,00	0,08	0,08	0,08	0,08	31,71	7,21	0,58	17,35	1,08	4,11	0,34	10,96	2,40
3	54,47	11,99	1,00	1,00	1,00	1,00	380,27	86,48	6,94	208,05	12,96	49,31	4,11	131,41	28,82
4	54,54	12,01	1,00	1,00	1,00	1,00	380,76	86,60	6,95	208,31	12,98	49,37	4,12	131,58	28,86
5	54,59	12,02	1,00	1,00	1,00	1,01	381,14	86,68	6,96	208,52	12,99	49,42	4,12	131,71	28,89
6	54,32	11,96	1,00	1,00	0,99	1,00	379,22	86,25	6,92	207,47	12,93	49,17	4,10	131,04	28,74
7	0,14	0,03	0,00	0,00	0,00	0,00	1,00	0,23	0,02	0,55	0,03	0,13	0,01	0,35	0,08
8	0,63	0,14	0,01	0,01	0,01	0,01	4,40	1,00	0,08	2,41	0,15	0,57	0,05	1,52	0,33
9	7,85	1,73	0,14	0,14	0,14	0,14	54,77	12,46	1,00	29,96	1,87	7,10	0,59	18,93	4,15
10	0,26	0,06	0,00	0,00	0,00	0,00	1,83	0,42	0,03	1,00	0,06	0,24	0,02	0,63	0,14
11	4,20	0,93	0,08	0,08	0,08	0,08	29,34	6,67	0,54	16,05	1,00	3,80	0,32	10,14	2,22
12	1,10	0,24	0,02	0,02	0,02	0,02	7,71	1,75	0,14	4,22	0,26	1,00	0,08	2,67	0,58
13	13,25	2,92	0,24	0,24	0,24	0,24	92,51	21,04	1,69	50,61	3,15	12,00	1,00	31,97	7,01
14	0,41	0,09	0,01	0,01	0,01	0,01	2,89	0,66	0,05	1,58	0,10	0,38	0,03	1,00	0,22
15	1,89	0,42	0,03	0,03	0,03	0,03	13,19	3,00	0,24	7,22	0,45	1,71	0,14	4,56	1,00
16	19,20	4,23	0,35	0,35	0,35	0,35	134,03	30,48	2,45	73,33	4,57	17,38	1,45	46,31	10,16

На основании критерия Фишера была выбрана линейная регрессия. Отношения первой к остальным дисперсиям больше 43, следовательно, сложность не константная.

Отношения второй к остальным дисперсиям больше 8, следовательно, сложность не логарифмическая.

Отношения третьей к остальным дисперсиям не превышает 1.26, из чего можно сделать вывод, что уравнение регрессии — третья. Усложнение моделей регрессии после третьей — незначимо.

Результат статистического эксперимента

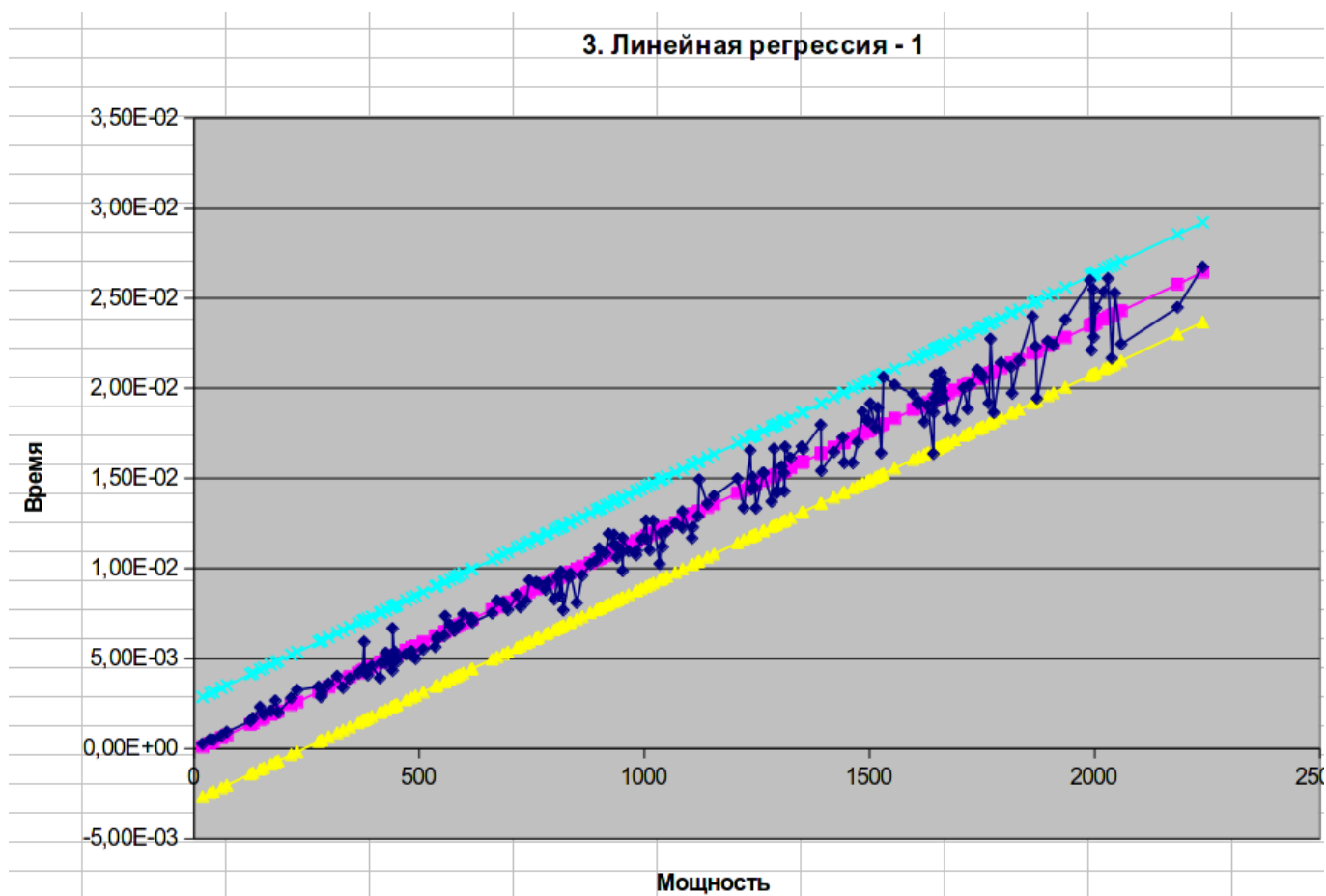


График подтверждает достоверность результатов обработки. Измеренные значения усреднены кривой регрессии, располагаясь примерно поровну выше и ниже её. Практически все отсчёты находятся в доверительном интервале $\pm 3 \sigma$, что соответствует допустимой ошибке 0,03% (нормальное распределение).

Для более сложных моделей наблюдается такая же картина, с примерно тем же размером доверительного интервала, что и означает бессмысленность усложнения модели.

Вывод

Ожидаемая временная сложность цепочки операций была $O(n \log n)$. Исходя из практической оценки, можно сделать вывод, что временная сложность линейная $O(n)$.

При выполнении курсовой работы были получены практические навыки в написании программ на языке C++ с вычислением практической временной сложности программы. В результате работы были получены данные, которые опровергли теоретическую оценку: по окончании 3-ей лабораторной работы был выдвинут тезис о сложности алгоритма $O(n \log n)$, однако в ходе курсовой выяснили, что временная сложность - линейная.

Список используемых источников

1. Колинко П.Г. Пользовательские структуры данных Часть 2: Методические указания к практическим занятиям на ПЭВМ и курсовому проектированию. / Методические указания по дисциплине «Алгоритмы и структуры данных» - Санкт-Петербург: СПбГЭТУ «ЛЭТИ», 2020.

Приложение

mycont.h — реализация контейнера множество+последовательность

main.cpp — демонстрационная программа

table.xls — таблица

in.txt — данные, полученные в ходе эксперимента

out.txt — данные, полученные программой RG41cb