

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 44**

Выполнил студенты гр.9308:

Дементьев Д.П., Ручкин Д.А.

Проверил:

Колинко 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 меняющейся мощностью, 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 общем случае временная сложность для произвольной вставки составляет $O(\log n)$ из-за необходимости находить место вставки проходом от корня дерева. При указании места вставки сложность будет составлять $O(1)$.

При вставки некоторого n количества значений, сложность будет равна $n \cdot \log(n)$

Пересечение множеств (operator&=)

Для пересечения по схеме слияния происходит не более $2 \cdot (N1 + N2) - 1$ сравнений, где $N1$ и $N2$ — размеры контейнеров, т.е. временная сложность алгоритма $O(n)$.

Объединение множеств (operator+=)

Для объединения по схеме слияния происходит не более $2 \cdot (N1 + N2) - 1$ сравнений, где $N1$ и $N2$ — размеры контейнеров, т.е. временная сложность алгоритма $O(n)$.

Разность множеств (operator-=)

Для разности по схеме слияния происходит не более $2 \cdot (N1 + N2) - 1$ сравнений, где $N1$ и $N2$ — размеры контейнеров, т.е. временная сложность алгоритма $O(n)$.

Симметрическая разность множеств (operator^=)

Для симметрической разности по схеме слияния происходит не более $2 \cdot (N1 + N2) - 1$ сравнений, где $N1$ и $N2$ — размеры контейнеров, т.е. временная сложность алгоритма $O(n)$.

Вырезание части последовательности(ERASE)

Временная сложность $O(n \log n)$ за счёт необходимости поддерживать согласованное содержимое дерева

Присоединение последовательностей (CONCAT)

Временная сложность $O(n \log n)$ за счёт необходимости поддерживать согласованное содержимое дерева

Замена элементов последовательности(CHANGE)

Временная сложность $O(n \log n)$ за счёт необходимости поддерживать согласованное содержимое дерева

4. Обработка результатов эксперимента

Для эксперимента была выполнено 201 итерация. Мощность множеств генерировалась от 1 до 2000. Для снижения количества дубликатов, размер универсума формировался с помощью домножения сгенерированной мощности на 3. Данные для эксперимента хранятся в файле in.txt

По данным измерениям был получен следующий результат работы программы RG41cb (находится в файле out.txt):

202 data pairs received												
Dispersion	RMS	k	C	log N	N	N log N	N^2	N^3	N^4	Steps	Code	
0.0151	0.123	1	0.198	0	0	0	0	0	0	202	++	
0.00466	0.0683	2	-0.509	0.109	0	0	0	0	0	138	++	
0.000431	0.0208	2	-0.0172	0	0.00025	0	0	0	0	140	--	
0.000398	0.02	3	0.0465	-0.0126	0.000271	0	0	0	0	329	--	
0.000378	0.0194	3	0.0105	0	-9.93e-005	4.57e-005	0	0	0	320	--	
0.000377	0.0194	4	-0.0189	0.00847	-0.000281	6.76e-005	0	0	0	625	--	
0.000768	0.0277	3	-0.0616	0	0.000335	0	-3.6e-008	0	0	163	**	
0.000497	0.0223	4	-0.0649	0	0.00163	-0.00022	1.75e-007	0	0	310	--	
0.000379	0.0195	5	-0.0137	0.00626	-0.000153	4.92e-005	9.91e-009	0	0	1121	--	
0.000639	0.0253	4	-0.0638	0	0.000453	0	-2.61e-007	9.79e-011	0	313	--	
0.000401	0.02	5	-0.0967	0.0263	-2.16e-005	0	2.17e-007	-5.5e-011	0	583	**	
0.000436	0.0209	5	-0.0673	0	0.00286	-0.000459	7.62e-007	-1.57e-010	0	586	**	
0.000395	0.0199	6	-0.114	0.0429	-0.00206	0.00033	-2.14e-007	3.03e-011	0	1089	--	
0.000519	0.0228	5	-0.0654	0	0.000661	0	-9.47e-007	7.76e-010	-2.06e-013	510	**	
0.000431	0.0208	6	-0.0689	0	0.00368	-0.000637	1.52e-006	-6.2e-010	1.11e-013	989	--	
0.000385	0.0196	7	-0.138	0.0688	-0.00656	0.00114	-2.21e-006	9.99e-010	-2.05e-013	1520	**	

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	16
1	1.00	0.31	0.03	0.03	0.03	0.02	0.05	0.03	0.03	0.04	0.03	0.03	0.03	0.03	0.03	0.03
2	3.24	1.00	0.09	0.09	0.08	0.08	0.16	0.11	0.08	0.14	0.09	0.09	0.08	0.11	0.09	0.08
3	35.02	10.82	1.00	0.92	0.88	0.87	1.78	1.15	0.88	1.48	0.93	1.01	0.92	1.20	1.00	0.89
4	37.89	11.70	1.08	1.00	0.95	0.95	1.93	1.25	0.95	1.60	1.01	1.10	0.99	1.30	1.08	0.97
5	39.91	12.33	1.14	1.05	1.00	1.00	2.03	1.31	1.00	1.69	1.06	1.15	1.05	1.37	1.14	1.02
6	40.03	12.37	1.14	1.06	1.00	1.00	2.04	1.32	1.00	1.69	1.06	1.16	1.05	1.38	1.14	1.02
7	19.64	6.07	0.56	0.52	0.49	0.49	1.00	0.65	0.49	0.83	0.52	0.57	0.51	0.68	0.56	0.50
8	30.38	9.38	0.87	0.80	0.76	0.76	1.55	1.00	0.76	1.29	0.81	0.88	0.80	1.04	0.87	0.77
9	39.84	12.31	1.14	1.05	1.00	1.00	2.03	1.31	1.00	1.69	1.06	1.15	1.04	1.37	1.14	1.02
10	23.62	7.30	0.67	0.62	0.59	0.59	1.20	0.78	0.59	1.00	0.63	0.68	0.62	0.81	0.68	0.60
11	37.65	11.63	1.08	0.99	0.94	0.94	1.92	1.24	0.95	1.59	1.00	1.09	0.99	1.29	1.08	0.96
12	34.59	10.68	0.99	0.91	0.87	0.86	1.76	1.14	0.87	1.46	0.92	1.00	0.91	1.19	0.99	0.88
13	38.18	11.79	1.09	1.01	0.96	0.95	1.94	1.26	0.96	1.62	1.01	1.10	1.00	1.31	1.09	0.97
14	29.10	8.99	0.83	0.77	0.73	0.73	1.48	0.96	0.73	1.23	0.77	0.84	0.76	1.00	0.83	0.74
15	34.98	10.81	1.00	0.92	0.88	0.87	1.78	1.15	0.88	1.48	0.93	1.01	0.92	1.20	1.00	0.89
16	39.23	12.12	1.12	1.04	0.98	0.98	2.00	1.29	0.98	1.66	1.04	1.13	1.03	1.35	1.12	1.00

При таком количестве экспериментов квантиль Фишера равен 1,26/1,39

Отношения первой к остальным дисперсиям больше 40, следовательно, сложность не константная.

Отношения второй к остальным дисперсиям больше 12, следовательно, сложность не логарифмическая.

Отношения третьей к остальным дисперсиям не превышает 1.26, из чего можно сделать вывод, что уравнение регрессии — третье.

Результат статистического эксперимента

3. Линейная регрессия - 1

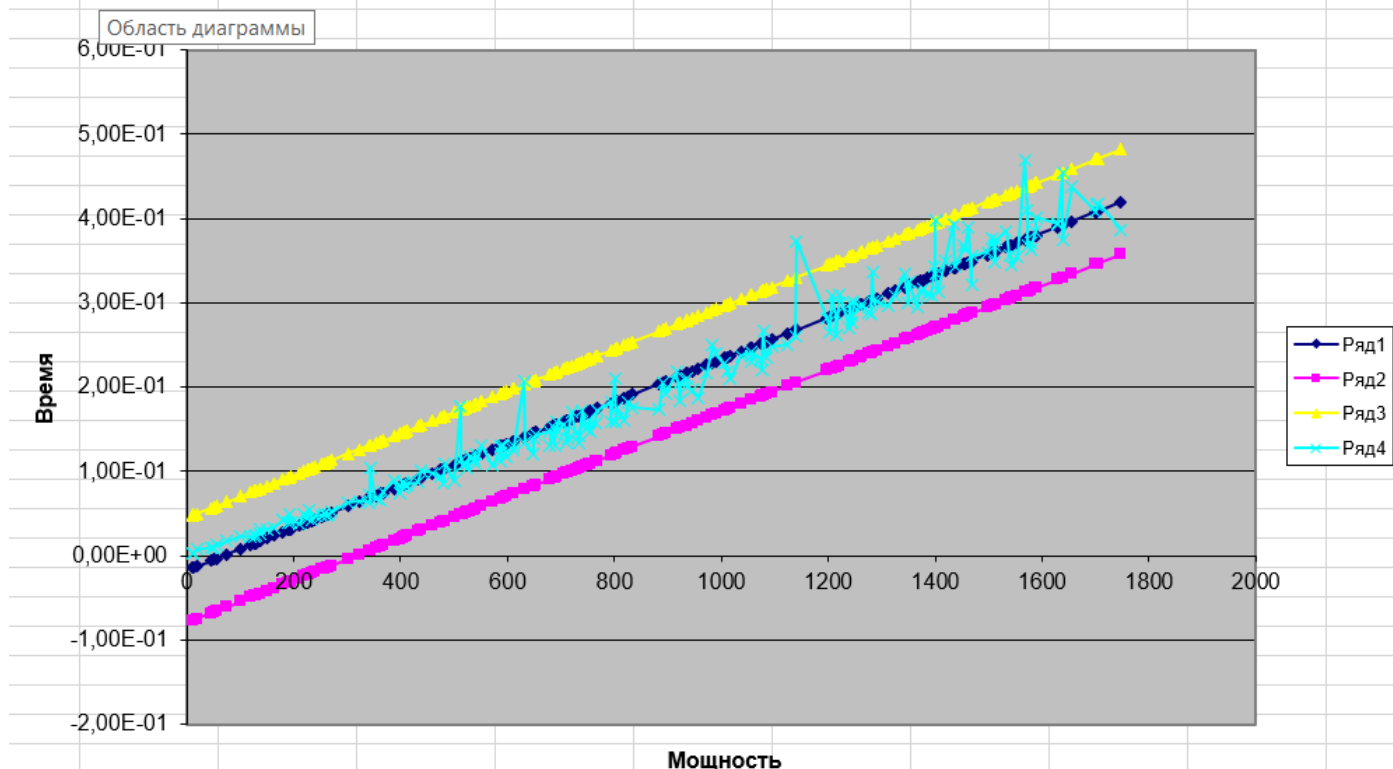


График подтверждает достоверность результатов обработки. Измеренные значения усреднены кривой регрессии, располагаясь примерно поровну выше и ниже её. Практически все отсчёты находятся в доверительном интервале $\pm 3\sigma$, что соответствует допустимой ошибке 0,03% (нормальное распределение).

Вывод

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

Некоторые возможные причины получения недостоверных результатов были приняты во внимание: проводился 201 опыт, чего более чем достаточно для усреднения значений; для оценки использовалась фактическая средняя мощность множеств, т.е. результаты эксперимента не получены со смещением; также отсутствуют данные, которые сильно отличаются от средних значений (вне доверительного интервала).

Возможная причина получения такого результата:

Вставка в дерево уже существующего ключа будет меньше $O(\log n)$ по причине использования отдельного поля, отвечающего за дубликаты. Для минимизации дубликатов было решено ввести зависимость максимального значения ключа от мощности генерируемого множества, но это дало аналогичные результаты.

При выполнении курсовой работы были получены практические навыки в написании программ на языке C++ с вычислением практической временной сложности программы. В результате работы теоретическая оценка сложности алгоритма $O(n \log n)$ не подтвердилась, и в ходе курсовой работы было получено, что временная сложность – линейная $O(n)$.

Список используемых источников

1. Колинъко 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

AVL.h — реализация АВЛ-дерева

set_seq.h — реализация контейнера

main.cpp — программа для эксперимента

in.txt — данные, полученные в ходе эксперимента

out.txt — данные, полученные программой RG41cb

table.xls — таблица для вычисления