Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и

информатики»

(СибГУТИ)

09.03.01 Информатика и вычислительная техника

Профиль: Электронно-вычислительные машины, комплексы, системы и сети

(очная форма обучения)

ОТЧЕТ ПО учебной ПРАКТИКЕ

на кафедре вычислительных систем

(наименование структурного подразделения СибГУТИ)

БИНАРНЫЙ ПОИСК. АНАЛИЗ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ

Выполнил:

студент института ИВТ

гр. ИВ-221 / /

«25» мая 2024 г. (подпись)

Проверил:

Руководитель от СибГУТИ / Перышкова Е.Н./

«25» мая 2024 г. (подпись)

Новосибирск 2024

**План-график проведения**  учебной **практики**

Вид практики

ГОРДОВА РОМАНА СЕРЕГЕЕВИЧА

Фамилия Имя Отчество студента

института Информатика и вычислительная техника , 2 курса,

гр. ИВ-221

Направление: 09.03.01 – Информатика и вычислительная техника

Код – Наименование направления (специальности)

Профиль: Электронно-вычислительные машины, комплексы, системы и сети

Место прохождения практики кафедра вычислительных систем

Объем практики: **108/3** часов/ЗЕ

Вид практики  ***учебная***

Тип практики ***ознакомительная***

Срок практики с "29" января 2024 г.

по "25" мая 2024 г.

Содержание практики\*:

|  |  |
| --- | --- |
| Наименование видов деятельности | Дата (начало – окончание) |
| 1. Общее ознакомление со структурным подразделением предприятия, вводный инструктаж по технике безопасности | 29.01.2024–01.02.2024 |
| 2. Выдача задания на практику, деление студентов на группы (если необходимо), определение конкретной индивидуальной темы, формирование плана работ | 02.02.2024–04.02.2024 |
| 3. Работа с библиотечными фондами структурного подразделения или предприятия, сбор и анализ материалов по теме практики | 06.02.2024–11.02.2024 |
| 4. Выполнение работ в соответствии с составленным планом:  –  –  –  (перечисление конкретных видов работ, связанных с выполнением поставленных задач) | 13.02.2024 – 20.05.2024 |
| 5. Анализ полученных результатов и произведенной работы  Составление отчета по практике, защита отчета | 22.05.2024–25.05.2024 |

\*В соответствии с программой практики

Руководитель от СибГУТИ / Перышкова Е.Н./

«29» \_\_\_\_01\_\_\_\_ 2024г. (подпись)

**ЗАДАНИЕ НА ПРАКТИКУ**

1. Реализовать функцию бинарного поиска на языке программирования C.

2. Провести тестирование алгоритма на различных входных данных, включая случаи успешного и неуспешного поиска.

3. Построить график.

4. Провести анализ вычислительной сложности бинарного поиска.

5. Сделать заключение.

**ВВЕДЕНИЕ**

В мире компьютерных наук и алгоритмов существует множество методов поиска, однако одним из наиболее эффективных и широко применяемых является бинарный поиск. Этот метод, иногда называемый методом дихотомии, представляет собой эффективный способ поиска элемента в упорядоченном массиве данных. Бинарный поиск основан на принципе деления пространства поиска на две части и последующем поиске в нужной половине.

Цель данной практической работы состоит в изучении и практическом применении бинарного поиска. В ходе работы мы также рассмотрим различные аспекты использования бинарного поиска, включая его временную сложность.

Понимание и умение применять бинарный поиск имеют важное значение для разработчиков программного обеспечения и специалистов в области информационных технологий, поскольку этот алгоритм является основой для решения множества задач, связанных с обработкой и анализом данных. Работа по изучению и практическому применению бинарного поиска позволит углубить знания о базовых алгоритмах и повысить навыки разработки программного обеспечения.

**ОСНОВНАЯ ЧАСТЬ**

Описание алгоритма:

1. На вход бинарному поиску передается упорядоченный массив, левая граница low, правая граница high и целевое значение x.

2. В начале выполнения алгоритма определяется середина массива mid.

3. Проверяется, равно ли значение в середине массива целевому значению. Если да, возвращается индекс этого элемента.

4. Если значение в середине больше целевого, поиск продолжается в левой половине массива, иначе - в правой.

5. Шаги 2-4 повторяются до тех пор, пока левая граница не станет больше правой (то есть, пока есть элементы для поиска).

6. Если элемент не найден после завершения цикла (то есть, левая граница стала больше правой), возвращается -1.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | **int** **binary\_search**(**int** \*array, **int** size\_of\_array, **int** x)  {  **int** low = **0**;  **int** high = size\_of\_array;  **while** (low <= high)  {  **int** mid = low + (high - low) / **2**;  **if** (array[mid] == x)  **return** mid;  **if** (array[mid] < x)  low = mid + **1**;  **else**  high = mid - **1**;  }  **return** -**1**;  } |

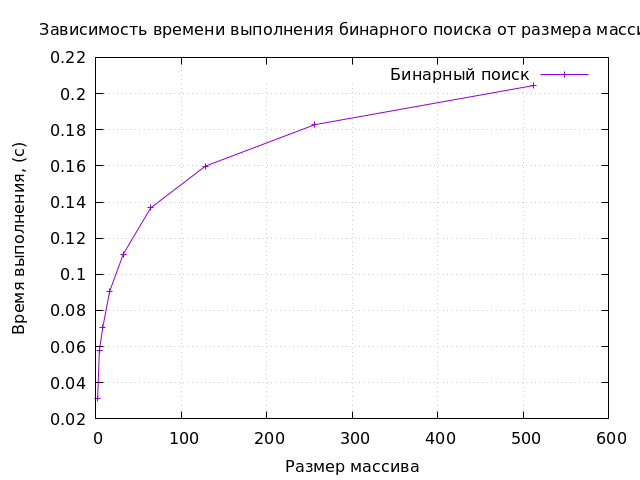
Для построения зависимости времени выполнения от размера входных данных обернем нашу процедуру в цикл, в котором мы будем увеличивать размер изначального массива.

Для уточнения результатов, будем искать случайный элемент в массиве миллион раз.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | **for** (**int** size\_of\_array = **2**; size\_of\_array <= **512**; size\_of\_array \*= **2**)  {  **int** array[size\_of\_array];  **for** (**int** i = **0**; i < size\_of\_array; i++)  {  array[i] = **10** \* i + (rand()) % **10**;  }  **double** time = -wtime();  **for** (**int** i = **0**; i < **1000000**; i++)  {  **int** need\_to\_find = array[rand() % (size\_of\_array - **1**)];  binary\_search(array, size\_of\_array, need\_to\_find);  }  time += wtime();  fprintf(fptr, "%d %f**\n**", size\_of\_array, time);  } |

Результат будем записывать в файл.

На основе выходных данным построим график зависимости времени от размера входных данных.



Алгоритм имеет логарифмическую сложность времени выполнения, так как на каждой итерации половина оставшихся элементов массива исключается из рассмотрения.

**ЗАКЛЮЧЕНИЕ**

В ходе данной практики мы успешно реализовали алгоритм бинарного поиска на языке программирования C и провели его тестирование. Анализируя зависимость времени выполнения алгоритма от размера входных данных, мы обнаружили, что бинарный поиск демонстрирует логарифмическую вычислительную сложность. Это означает, что время выполнения алгоритма растет медленно при увеличении размера входных данных, что делает его эффективным для поиска в больших объемах данных.

Полученные результаты подтверждают важность использования бинарного поиска в практических задачах, особенно в случаях, когда требуется быстрый и эффективный поиск в упорядоченных данных. Бинарный поиск может быть применен в различных областях, включая поиск элементов в массивах, базах данных, а также в алгоритмах оптимизации.

В дальнейшем исследовании можно рассмотреть дополнительные аспекты бинарного поиска, такие как его сравнение с другими алгоритмами поиска, оптимизация его реализации и адаптация для конкретных задач. Это позволит более глубоко понять принципы работы алгоритма и расширить его применимость в различных областях информационных технологий.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Kernighan, B., & Ritchie, D. M. (1988). The C Programming Language (2nd ed.). Prentice Hall.
2. Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Addison-Wesley.
3. GeeksforGeeks. (https://www.geeksforgeeks.org/)

**ПРИЛОЖЕНИЯ**

**main.c**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66 | #include <stdio.h>  #include <time.h>  #include <time.h>  #include <stdlib.h>  **void** **print\_array**(**int** \*array, **int** n)  {  **for** (**int** i = **0**; i < n; i++)  {  printf("%d ", array[i]);  }  printf("**\n**");  }  **double** **wtime**()  {  **struct** timespec ts;  clock\_gettime(CLOCK\_MONOTONIC, &ts);  **return** ts.tv\_sec + ts.tv\_nsec \* **1E-9**;  }  **int** **binary\_search**(**int** \*array, **int** size\_of\_array, **int** x)  {  **int** low = **0**;  **int** high = size\_of\_array;  **while** (low <= high)  {  **int** mid = low + (high - low) / **2**;  **if** (array[mid] == x)  **return** mid;  **if** (array[mid] < x)  low = mid + **1**;  **else**  high = mid - **1**;  }  **return** -**1**;  }  **int** **main**()  {  srand(time(NULL));  **FILE** \*fptr;  fptr = fopen("data.dat", "w");  **for** (**int** size\_of\_array = **2**; size\_of\_array <= **512**; size\_of\_array \*= **2**)  {  **int** array[size\_of\_array];  **for** (**int** i = **0**; i < size\_of\_array; i++)  {  array[i] = **10** \* i + (rand()) % **10**;  // printf("%d ", array[i]);  }  **double** time = -wtime();  **for** (**int** i = **0**; i < **1000000**; i++)  {  **int** need\_to\_find = array[rand() % (size\_of\_array - **1**)];  binary\_search(array, size\_of\_array, need\_to\_find);  }  time += wtime();  fprintf(fptr, "%d %f**\n**", size\_of\_array, time);  }  fclose(fptr);  **return** **0**; |

plot.gp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | set terminal pngcairo enhanced font 'Verdana,**12**'  set output 'binary\_search\_plot.png'  set title "Зависимость времени выполнения бинарного поиска от размера массива"  set xlabel "Размер массива"  set ylabel "Время выполнения, (c)"  set grid  plot "data.dat" using **1**:**2** with linespoints title "Бинарный поиск" |

Makefile

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | CC = gcc  CFLAGS = -Wall -Wextra  TARGET = binary\_search  SRCDIR = src  **all:** run  $(TARGET): $(SRCDIR)/main.c  $(CC) $(CFLAGS) -o $(TARGET) $(SRCDIR)/main.c  **run:** $(TARGET)  ./$(TARGET)  gnuplot $(SRCDIR)/plot.gp  **clean:**  rm -f $(TARGET) data.dat binary\_search\_plot.png |

**Отзыв о работе студента**

|  |
| --- |
|  |
| (ФИО студента) |
|  |

Уровень освоения компетенций

|  |  |  |
| --- | --- | --- |
|  |  | |
|  | (ФИО студента) | |
|  |  | |
| Компетенции | | Уровень сформированности  компетенций |
| *ОПК-2 - Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решения задач профессиональной деятельности* | |  |

отметка о зачете с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики от СибГУТИ:

Заведующий кафедрой ВС Перышкова Евгения Николаевна

Должность руководителя подпись ФИО руководителя

"25" мая 2024 г.