ADSA PROJECT
College Library books Arrangement

Group-7:
Parupalli Guna Gokul(AP18110010279)
Nuthalapati Sai Kumar(AP18110010283)
Lakshmi Pranav Kakumanu (AP18110010285)
Annamaneni Ajay Chowdari (AP18110010287)
Chirumamilla Prudhvi Chowdary (AP18110010288)

Objective:
The main objective of this project is to find out the best sorting algorithm to sort a set of

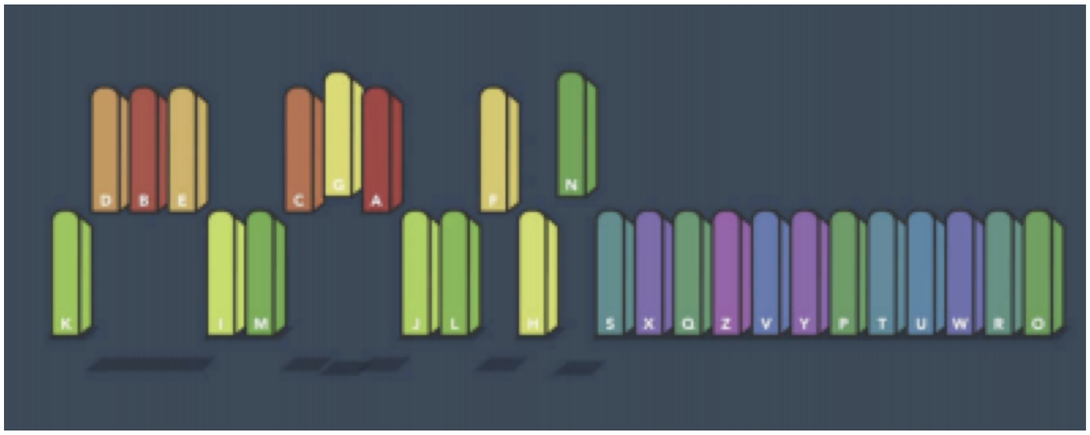
strings in a file and to understand how it works and why it is best out of others. We also concentrate on the time and space complexities of the algorithm that we use.

Problem Statement: Suppose you work at a college library. You are in the middle of

G
C D E I K M S X A Q Z J N V Y P L T U
WHAT'S THE FASTEST WAY TO ALPHABETIZE YOUR BOOKSHELF?

quiet afternoon when suddenly a shipment of 3928 different books arrives.

The books have been dropped off in one long straight line, but they are all out of order, and the automatic sorting system is broken. To make matters worse, classes will start tomorrow, which means that first thing in the morning, students will show up in droves looking for these books.



How can you get them all sorted in time?

Simulate the given scenario using C code.

Complexity Analysis: Perform the algorithmic time complexity analysis for the solution you propose. Also give the space complexity.

Outcomes: You must be in a position to identify the concepts and ideas – C programming constructs used to solve real-life problems.

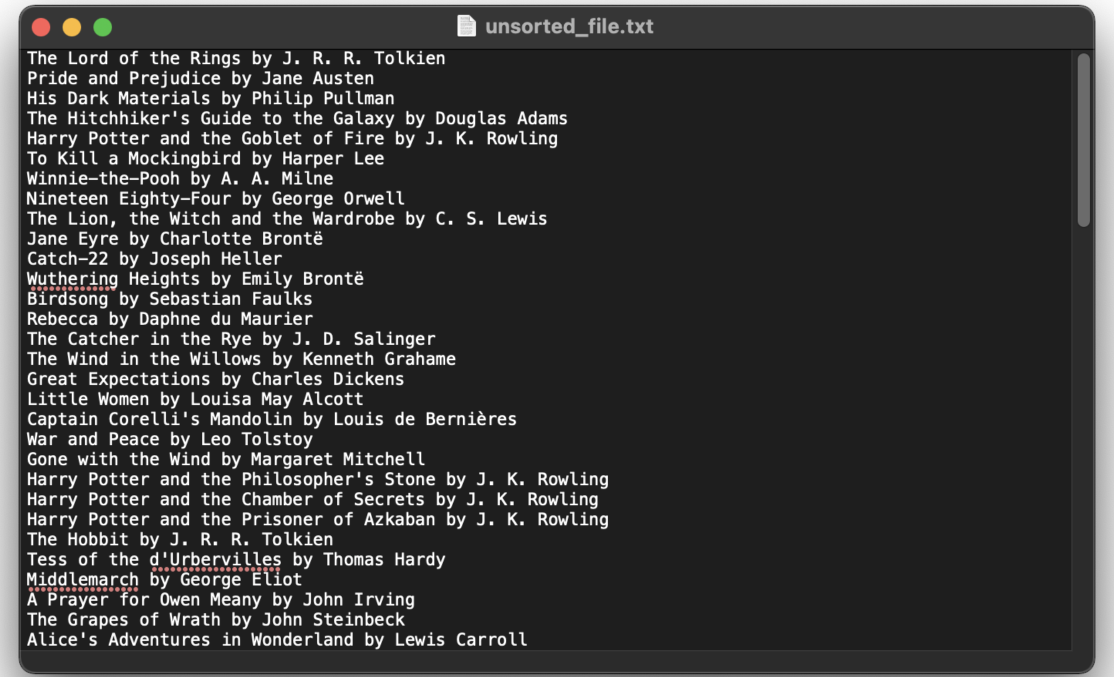
Steps:

1. Consider a text file containing titles of books.
2. If the file is empty return the error message
3. If the file is not empty, read the file line by line till it reaches the end
4. Quick sort:.
5. Step 1 - Consider the middle element (i.e., middle line of the file) of the array as pivot.
6. Step 2 - Define two variables i and j. Set i and j to the first and last elements of the array (i.e., first and last lines of the file) respectively.
7. Step 3 - Increment i until array[i] > pivot then stop.
8. Step 4 - Decrement j until array[j] < pivot then stop.
9. Step 5 - If i < j then swap array[i] and array[j].
10. Step 6 - Repeat steps 3,4 & 5 until i > j.
11. Step 8 − if left ≥ right, the point where they met is the new pivot.

Code:

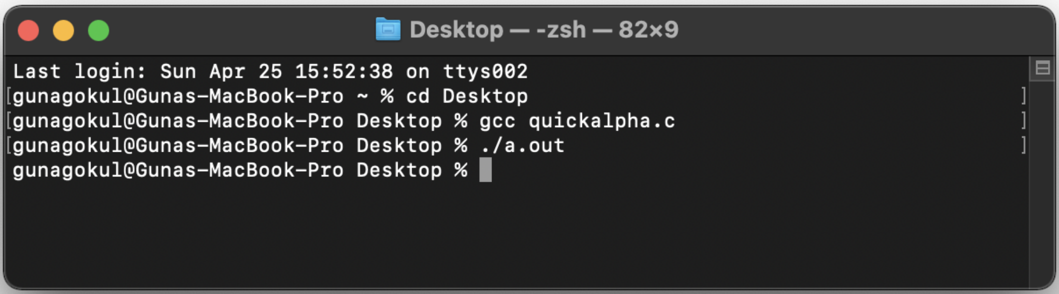
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void quicksort(char **array, int first_line, int last_line) ;

int main(int argc, char* argv[])
{
   int i = 0 ;
   int j = 0 ;
   int line_count = 0 ;
   int line_number = 300 ;
   char *lines[100] ;
   FILE *foutptr ;
   FILE *finptr ;

   finptr = fopen("sorted_file.txt", "r") ;
   if(finptr == NULL)
   {
      perror("ERROR: input file pointer") ;
      return 1 ;
   }

   foutptr = fopen("sorted.txt", "w") ;
   if(foutptr == NULL)
   {
      perror("ERROR: output file pointer") ;
      return 1 ;
   }

   lines[i] = malloc(BUFSIZ) ;
   while(fgets(lines[i],BUFSIZ,finptr) && i < line_number)                    //O(n)
   {
      i++ ;
      lines[i] = malloc(BUFSIZ) ;
   }

   line_count = i ;
   i = 0 ;
   char **all_lines = lines ;

   quicksort(all_lines, 0, line_count-1) ;

   for(j = 0 ; j < line_count ; j++)                                          //O(n)
   {
      fprintf(foutptr, "%s", lines[j]) ;
   }

   return 0 ;
}

void quicksort(char **array, int first_line, int last_line)
{
   int i = 0, j = 0, x = 0i, p = (first_line + last_line)/2 ;
   char pivot[BUFSIZ] ;
   char *temp_ptr ;

   strcpy(pivot,array[p]) ;

   i = first_line ;
   j = last_line ;

   if(last_line - first_line < 1)
   {
      return ;
   }


   while(i <= j)                                                     //  O(logn)
   {
      while(strcmp(array[i], pivot) < 0 && i < last_line)          // O(n)
      {
          i++ ;
      }
      while(strcmp(array[j], pivot) > 0 && j > first_line)         //O(n)
      {
          j-- ;
      }
      if(i <= j)
      {
          temp_ptr = array[i] ;
          array[i] = array[j] ;
          array[j] = temp_ptr ;
          i++ ;
          j-- ;
      }
   }

   if(first_line < j)
   {
      quicksort(array, first_line, j) ;
   }

   x = 0 ;

   if(i < last_line)
   {
      quicksort(array, i, last_line) ;
   }

   x = 0 ;
   return ;
}
```

Input:

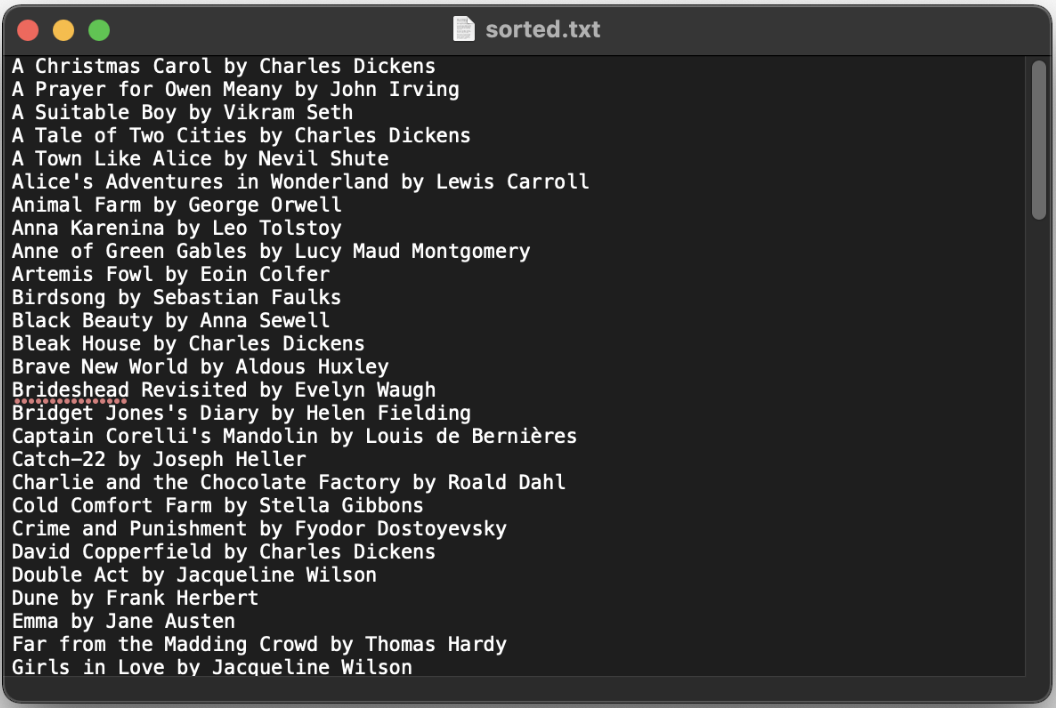


```
The Lord of the Rings by J. R. R. Tolkien
Pride and Prejudice by Jane Austen
His Dark Materials by Philip Pullman
The Hitchhiker's Guide to the Galaxy by Douglas Adams
Harry Potter and the Goblet of Fire by J. K. Rowling
To Kill a Mockingbird by Harper Lee
Winnie-the-Pooh by A. A. Milne
Nineteen Eighty-Four by George Orwell
The Lion, the Witch and the Wardrobe by C. S. Lewis
Jane Eyre by Charlotte Brontë
Catch-22 by Joseph Heller
Wuthering Heights by Emily Brontë
Birdsong by Sebastian Faulks
Rebecca by Daphne du Maurier
The Catcher in the Rye by J. D. Salinger
The Wind in the Willows by Kenneth Grahame
Great Expectations by Charles Dickens
Little Women by Louisa May Alcott
Captain Corelli's Mandolin by Louis de Bernières
War and Peace by Leo Tolstoy
Gone with the Wind by Margaret Mitchell
Harry Potter and the Philosopher's Stone by J. K. Rowling
Harry Potter and the Chamber of Secrets by J. K. Rowling
Harry Potter and the Prisoner of Azkaban by J. K. Rowling
The Hobbit by J. R. R. Tolkien
Tess of the d'Urbervilles by Thomas Hardy
Middlemarch by George Eliot
A Prayer for Owen Meany by John Irving
The Grapes of Wrath by John Steinbeck
Alice's Adventures in Wonderland by Lewis Carroll
```

Output:



```
Last login: Sun Apr 25 15:52:38 on ttys002
gunagokul@Gunas-MacBook-Pro ~ % cd Desktop
gunagokul@Gunas-MacBook-Pro Desktop % gcc quickalpha.c
gunagokul@Gunas-MacBook-Pro Desktop % ./a.out
gunagokul@Gunas-MacBook-Pro Desktop %
```



```
A Christmas Carol by Charles Dickens
A Prayer for Owen Meany by John Irving
A Suitable Boy by Vikram Seth
A Tale of Two Cities by Charles Dickens
A Town Like Alice by Nevil Shute
Alice's Adventures in Wonderland by Lewis Carroll
Animal Farm by George Orwell
Anna Karenina by Leo Tolstoy
Anne of Green Gables by Lucy Maud Montgomery
Artemis Fowl by Eoin Colfer
Birdsong by Sebastian Faulks
Black Beauty by Anna Sewell
Bleak House by Charles Dickens
Brave New World by Aldous Huxley
Brideshead Revisited by Evelyn Waugh
Bridget Jones's Diary by Helen Fielding
Captain Corelli's Mandolin by Louis de Bernières
Catch-22 by Joseph Heller
Charlie and the Chocolate Factory by Roald Dahl
Cold Comfort Farm by Stella Gibbons
Crime and Punishment by Fyodor Dostoyevsky
David Copperfield by Charles Dickens
Double Act by Jacqueline Wilson
Dune by Frank Herbert
Emma by Jane Austen
Far from the Madding Crowd by Thomas Hardy
Girls in Love by Jacqueline Wilson
```

<u>Overall Time Complexity and Space Complexity:</u>

Time complexity:

$$O(n)+O(n)+(O(\log n)*(O(n)+O(n)))$$

$$=2*O(n)+O(\log n)*2*O(n)$$

$$=O(n)+O(\log n)*O(n)$$

$$=O(n)+O(n\log n)$$

$$=O(n\log n)$$

Space complexity: $O(n\log n)$

Problems Faced:

- ➢ We faced a problem in finding out which Divide and conquer sorting algorithm is best suited for it.
- ➢ As there are 2 different types of quick sorting algorithms we felt confused to choose one among them.
- ➢ As we have to sort the books based on alphabetical order of their names since we do not give any serial numbers to them, it was difficult to apply quicksort.

Conclusion:

We found out the time and space complexities of the best suiting algorithm for sorting the books in a library.

Finally, we conclude that Quicksort is the best sorting algorithm for this project.

Some Real Life Applications of Quick Sort:

- ➢ It can be used in Filters in E-Commerce Websites.
- ➢ This algorithm can be used to sort any text file of strings.
- ➢ It is used in various government and private organizations for the purpose of sorting various data like sorting of accounts/profiles by name or any given ID, sorting transactions by time or locations, sorting files by name or date of creation etc.