\documentclass[14pt,a4paper,report]{ncc}

\usepackage[a4paper, mag=1000, left=2.5cm, right=1cm, top=2cm, bottom=2cm, headsep=0.7cm, footskip=1cm]{geometry}

\usepackage[utf8]{inputenc}

\usepackage[english,russian]{babel}

\usepackage{indentfirst}

\usepackage[dvipsnames]{xcolor}

\usepackage[colorlinks]{hyperref}

\usepackage{listings}

\usepackage{caption}

\usepackage{graphicx}

\DeclareCaptionFont{white}{\color{white}}

\DeclareCaptionFormat{listing}{\colorbox{gray}{\parbox{\textwidth}{#1#2#3}}}

\captionsetup[lstlisting]{format=listing,labelfont=white,textfont=white}

\lstset{% Собственно настройки вида листинга

inputencoding=utf8, extendedchars=\true, keepspaces = true, % поддержка кириллицы и пробелов в комментариях

}

\begin{document}

% Переоформление некоторых стандартных названий

\renewcommand{\chaptername}{Практика}

% Оформление титульного листа

\begin{titlepage}

\begin{center}

\textsc « МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ»\\[5mm]

Федеральное государственное бюджетное образовательное учреждение высшего образования

АДЫГЕЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ\\[2mm]

Инженерно-физический факультет

Кафедра автоматизированных систем обработки информации и управления}

\vfill

\textbf{Отчет по практике\\[3mm]

«Сортировка Быстрая и Слиянием.»\\[6mm]

Вариант 4

\\[20mm]

}

\end{center}

\hfill

\begin{minipage}{.5\textwidth}

2 курс, группа 2ИВТ1\\[2mm]

Выполнил:\\

О.О. Богомолов 2024 г\\[5mm]

Руководитель:\\[2mm]

С.В. Теплоухов 2024 г\\

\end{minipage}%

\vfill

\begin{center}

Майкоп, \theyear\ г.

\end{center}

\end{titlepage}

\begin{document}

\section {Введение}

\label{{sec:intro}}

% Что должно быть во введении

\begin{enumerate}

\item Текстовая формулировка задачи

\item Код данной задачи

\item Скриншот программы

\end{enumerate}

\section {Вариант 4}

\subsection{Задание}

\item Сортировки Быстрая и Слиянием..

\subsection{Теория}

\item \textbf{Быстрая сортировка}, хоть и была разработана более 40 лет назад, является наиболее широко применяемым и одним их самых эффективных алгоритмов. Метод основан на подходе "разделяй-и-властвуй". Общая схема такова:

\begin{itemize}

\item из массива выбирается некоторый опорный элемент a[i],

\end{itemize}

\begin{itemize}

\item запускается процедура разделения массива, которая перемещает все ключи, меньшие, либо равные a[i], влево от него, а все ключи, большие, либо равные a[i] - вправо,

\end{itemize}

\begin{itemize}

\item теперь массив состоит из двух подмножеств, причем левое меньше, либо равно правого,

\end{itemize}

\begin{itemize}

\item для обоих подмассивов: если в подмассиве более двух элементов, рекурсивно запускаем для него ту же процедуру.

\end{itemize}

В конце получится полностью отсортированная последовательность.

Разделение массива:

На входе массив a[0]...a[N] и опорный элемент p, по которому будет производиться разделение.

\begin{itemize}

\item Введем два указателя: i и j. В начале алгоритма они указывают, соответственно, на левый и правый конец последовательности.

\end{itemize}

\begin{itemize}

\item Будем двигать указатель i с шагом в 1 элемент по направлению к концу массива, пока не будет найден элемент a[i] >= p. Затем аналогичным образом начнем двигать указатель j от конца массива к началу, пока не будет найден a[j] <= p.

\end{itemize}

\begin{itemize}

\item Далее, если i <= j, меняем a[i] и a[j] местами и продолжаем двигать i,j по тем же правилам...

\end{itemize}

\begin{itemize}

\item Повторяем шаг 3, пока i <= j.

\end{itemize}

Рассмотрим работу процедуры для массива a[0]...a[6] и опорного элемента p = a[3].

\item В простейшем случае алгоритм выглядит так:

\includegraphics[width = 15cm]{algoritm1.PNG}

\item Теперь массив разделен на две части: все элементы левой меньше либо равны p, все элементы правой - больше, либо равны p. Разделение завершено.

\\[15м]

\item \textbf{Сортировка слиянием} также построена на принципе "разделяй-и-властвуй", однако реализует его несколько по-другому, нежели quickSort. А именно, вместо разделения по опорному элементу массив просто делится пополам.

Функция merge на месте двух упорядоченных массивов a[lb]...a[split] и a[split+1]...a[ub] создает единый упорядоченный массив a[lb]...a[ub].

Пример работы алгоритма на массиве 3 7 8 2 4 6 1 5..

\\[5 mm]

\includegraphics[width = 15cm]{algoritm2.PNG}

\item Рекурсивный алгоритм обходит получившееся дерево слияния в прямом порядке. Каждый уровень представляет собой проход сортировки слияния - операцию, полностью переписывающую массив.

Обратим внимание, что деление происходит до массива из единственного элемента. Такой массив можно считать упорядоченным, а значит, задача сводится к написанию функции слияния merge.

Один из способов состоит в слиянии двух упорядоченных последовательностей при помощи вспомогательного буфера, равного по размеру общему количеству имеющихся в них элементов. Элементы последовательностей будут перемещаться в этот буфер по одному за шаг.

merge ( упорядоченные последовательности A, B , буфер C ) {

пока A и B непусты {

cравнить первые элементы A и B

переместить наименьший в буфер

}

если в одной из последовательностей еще есть элементы

дописать их в конец буфера, сохраняя имеющийся порядок

}

Пример работы на последовательностях 2 3 6 7 и 1 4 5

\includegraphics[width = 15cm]{algoritm3.PNG}

\item Результатом является упорядоченная последовательность, находящаяся в буфере. Каждая операция слияния требует n пересылок и n сравнений, где n - общее число элементов, так что время слияния: Theta(n).

\section{Ход работы}

\label{sec:exp}

\subsection{Код программы}

\label{sec:exp:code}

\begin{verbatim}

Метод быстрой сортировки

#include <iostream>

#include <ctime>

using namespace std;

const int n = 20;

void quicksortHoara(int\* M, int l, int r)

{

int i = l, j = r;

double count, x;

x = M[(l + r) / 2];

do

{

while (M[i] < x) i++;

while (M[j] > x) j--;

if (i <= j)

{

count = M[i];

M[i] = M[j];

M[j] = count;

i++; j--;

}

} while (i <= j);

if (i < r)

quicksortHoara(M, i, r);

if (j > l)

quicksortHoara(M, l, j);

}

//главная функция

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

int A[n], i; //объявление динамического массива

cout << "Вывод массива:\n";

for (i = 0; i < n; i++) //ввод массива

{

A[i] = rand() % 100;

cout << A[i] << " ";

}

cout << endl;

quicksortHoara(A, 0, n - 1);

cout << "Отсортированный массив:\n";

for (i = 0; i < n; i++) //ввод массива

{

cout << A[i] << " ";

}

cout << endl;

system("pause");

return 0;

}

Сортировка Методом слияния

#include <iostream>

using namespace std;

//функция, сливающая массивы

void Merge(int\* A, int first, int last)

{

int middle, start, final, j;

int\* mas = new int[100];

middle = (first + last) / 2; //вычисление среднего элемента

start = first; //начало левой части

final = middle + 1; //начало правой части

for (j = first; j <= last; j++) //выполнять от начала до конца

if ((start <= middle) && ((final > last) || (A[start] < A[final])))

{

mas[j] = A[start];

start++;

}

else

{

mas[j] = A[final];

final++;

}

//возвращение результата в список

for (j = first; j <= last; j++) A[j] = mas[j];

delete[]mas;

};

//рекурсивная процедура сортировки

void MergeSort(int\* A, int first, int last)

{

{

if (first < last)

{

MergeSort(A, first, (first + last) / 2); //сортировка левой части

MergeSort(A, (first + last) / 2 + 1, last); //сортировка правой части

Merge(A, first, last); //слияние двух частей

}

}

};

//главная функция

void main()

{

setlocale(LC\_ALL, "Rus");

int i, n;

int\* A = new int[100];

cout << "Размер массива > "; cin >> n;

for (i = 1; i <= n; i++)

{

cout << i << " элемент > "; cin >> A[i];

}

MergeSort(A, 1, n); //вызов сортирующей процедуры

cout << "Упорядоченный массив: "; //вывод упорядоченного массива

for (i = 1; i <= n; i++) cout << A[i] << " ";

delete[]A;

system("pause>>void");

}

\end{verbatim}

\begin{document}

\section{Скриншот программы }

Сортировки Быстрая и Слиянием.

\label{sec:picexample}

\begin{figure}[h]

\centering

\includegraphics[scale=0.8]{sortirovka1.PNG}

\\[5 mm]

\caption{Скриншот программы метод быстрой сортировки}\label{fig:par}

\\[5 mm]

\includegraphics[scale=0.8]{sortirovka2.PNG}

\end{figure}

\newpage

\begin{figure}[h]

\caption{Скриншот программы сортивка методом слиянием }\label{fig:par}

\end{figure}

\section{ Библиографические ссылки}

Для изучения «внутренностей» \TeX{} необходимо

изучить~\cite{Knuth-2003}, а для использования \LaTeX{} лучше

почитать~\cite{Lvovsky-2003, Voroncov-2005}.

\begin{thebibliography}{9}

\bibitem{Knuth-2003}Кнут Д.Э. Всё про \TeX. \newblock —- Москва: Изд. Вильямс, 2003 г. 550~с.

\bibitem{Lvovsky-2003}Львовский С.М. Набор и верстка в системе \LaTeX{}. \newblock —- 3-е издание, исправленное и дополненное, 2003 г.

\bibitem{Voroncov-2005}Воронцов К.В. \LaTeX{} в примерах. 2005 г.

\end{thebibliography}

\end{document}

\end{document}

\end{document}

\end{document}