**Pengimplementasian Selection Sort**

**Source code :**

/\*

Nama File : tugasSelectionSort.c

Programmer : Manja Fani Oktavia

NIM : 22343056

\*/

#include <stdio.h>

void selectionSort(int arr[], int n){

int i, j, min\_idx;

for (i = 0; i < n-1; i++){

min\_idx = i;

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

if (arr[j] < arr[min\_idx])

min\_idx = j;

int temp = arr[min\_idx];

arr[min\_idx] = arr[i];

arr[i] = temp;

}

}

int main(){

int arr[] = {64, 25, 12, 22, 11};

int n = sizeof(arr)/sizeof(arr[0]);

selectionSort(arr, n);

printf("Sorted array: \n");

for (int i=0; i < n; i++)

printf("%d ", arr[i]);

return 0;

}

**Prinsip kerja selection sort :**

1. Pada setiap iterasi, elemen terkecil di antara elemen yang belum diurutkan dicari dengan mencari indeks elemen terkecil di dalam array.
2. Elemen terkecil tersebut kemudian ditukar dengan elemen pertama di dalam array yang belum diurutkan.
3. Proses ini diulangi untuk setiap elemen yang belum diurutkan, sehingga array diurutkan secara keseluruhan.

**Pengimplementasian Merge Sort**

**Source code :**

/\*

Nama File : tugasMergeSort.c

Programmer : Manja Fani Oktavia

NIM : 22343056

\*/

#include <stdio.h>

void merge(int arr[], int l, int m, int r)

{

int i, j, k;

int n1 = m - l + 1;

int n2 = r - m;

int L[n1], R[n2];

for (i = 0; i < n1; i++)

L[i] = arr[l + i];

for (j = 0; j < n2; j++)

R[j] = arr[m + 1+ j];

i = 0;

j = 0;

k = l;

while (i < n1 && j < n2)

{

if (L[i] <= R[j])

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

void mergeSort(int arr[], int l, int r)

{

if (l < r)

{

int m = l+(r-l)/2;

mergeSort(arr, l, m);

mergeSort(arr, m+1, r);

merge(arr, l, m, r);

}

}

int main()

{

int arr[] = {12, 11, 13, 5, 6, 7};

int arr\_size = sizeof(arr)/sizeof(arr[0]);

mergeSort(arr, 0, arr\_size - 1);

printf("Sorted array: \n");

for (int i=0; i < arr\_size; i++)

printf("%d ", arr[i]);

return 0;

}

**Prinsip kerja merge sort :**

1. Prinsip kerja Merge Sort adalah dengan membagi array yang besar menjadi dua bagian yang lebih kecil secara rekursif hingga tersisa satu elemen saja atau array tersebut dianggap terurut. Kemudian, dua bagian tersebut akan digabungkan secara terurut hingga membentuk satu array besar yang terurut.
2. Proses pengurutan dilakukan dalam fungsi merge(), yang menggabungkan dua bagian array terurut menjadi satu bagian array terurut. Fungsi merge() ini akan membandingkan dua elemen array dan memilih elemen yang lebih kecil untuk dimasukkan ke dalam array hasil penggabungan. Fungsi ini akan diulang sampai seluruh elemen dalam dua array tergabung.
3. Proses rekursif terjadi dalam fungsi mergeSort(), dimana array dibagi menjadi dua bagian sama besar hingga satu elemen saja. Kemudian, dua bagian tersebut akan digabungkan secara terurut hingga menjadi satu array besar yang terurut.
4. Dengan prinsip kerja seperti ini, Merge Sort menjadi algoritma sorting yang cepat dan efisien dengan kompleksitas waktu O(n log n). Meskipun demikian, penggunaan memori yang besar dalam pembuatan array sementara saat penggabungan menjadi satu array besar menjadi kekurangan dari Merge Sort.

Ada banyak aplikasi yang mengimplementasikan selection sort dan merge sort. Di bawah ini, saya berikan beberapa contoh:

1. Aplikasi Spreadsheet: Spreadsheet seperti Microsoft Excel menggunakan algoritma sorting untuk menyusun data dalam urutan yang sesuai. Dalam hal ini, aplikasi tersebut dapat menggunakan selection sort atau merge sort, tergantung pada data yang akan diurutkan.
2. Aplikasi Database: Database seperti MySQL dan PostgreSQL menggunakan algoritma sorting untuk mengurutkan data dalam tabel. Dalam hal ini, aplikasi tersebut dapat menggunakan selection sort atau merge sort, tergantung pada data yang akan diurutkan.
3. Aplikasi Permainan: Beberapa permainan memerlukan pengurutan data, seperti pengurutan daftar pemain berdasarkan skor tertinggi. Dalam hal ini, permainan tersebut dapat menggunakan selection sort atau merge sort untuk mengurutkan data.
4. Aplikasi Penjualan Online: Aplikasi penjualan online seperti Amazon dan eBay menggunakan sorting untuk menyajikan produk-produk dalam urutan yang sesuai. Dalam hal ini, aplikasi tersebut dapat menggunakan selection sort atau merge sort, tergantung pada data yang akan diurutkan.
5. Aplikasi Web: Aplikasi web seperti Google menggunakan sorting untuk mengurutkan hasil pencarian dalam urutan yang sesuai. Dalam hal ini, aplikasi tersebut dapat menggunakan selection sort atau merge sort, tergantung pada data yang akan diurutkan.