Laporan Jobsheet 9

Dosen pengampu : Randi Proska Sandra, M.Sc

Kode Kelas : 202323430158

Disusun Oleh :

Fhandy Nofalino Akhsan
23343065

PROGRAM STUDI INFORMATIKA (NK)
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG
2024

Pengantar

Mata kuliah Praktek Struktur Data merupakan salah satu mata kuliah yang wajib diambil oleh mahasiswa teknik informatika. Dalam mata kuliah ini, kita akan belajar mengenai konsep-konsep dasar dari struktur data, seperti pointer, struct, dan array. Selain itu, kita juga akan mempelajari tentang beberapa jenis linked list, yaitu link list, double link list, dancircular link list. Berikut adalah ringkasan mengenai materi-materi tersebut.

Pointer, Struct, dan Array: Pada dasarnya, pointer, struct, dan array adalah konsep-konsep dasar dari bahasa pemrograman C. Pada praktik struktur data, kita akan mempelajari cara menggunakannya dalam membuat struktur data yang lebih kompleks.

1. **Selection Sort**

Selection Sort bekerja dengan cara menemukan elemen terkecil dalam array dan menukarnya dengan elemen pertama. Kemudian, elemen terkecil berikutnya dicari dari sisa array dan ditukar dengan elemen kedua, dan seterusnya. Proses ini terus berulang sampai seluruh array terurut.

Source Code

```
//created by Fhandy Nofalino Akhsan 23343065


#include <stdio.h>


// Fungsi untuk menukar dua angka
void swap(int *xp, int *yp) {
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}


// Fungsi untuk mengimplementasikan Selection Sort
void selectionSort(int arr[], int n) {
    int i, j, min_idx;


    // Memindahkan batas dari array yang tidak terurut
    for (i = 0; i < n-1; i++) {
        // Menemukan elemen terkecil di array yang tidak terurut
        min_idx = i;
        for (j = i+1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;
```

```
        // Menukar elemen terkecil dengan elemen pertama
        swap(&arr[min_idx], &arr[i]);
    }
}

// Fungsi untuk mencetak array
void printArray(int arr[], int size) {
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

// Fungsi utama
int main() {
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr)/sizeof(arr[0]);
    printf("Array sebelum disorting: \n");
    printArray(arr, n);

    selectionSort(arr, n);
    printf("Array setelah disorting dengan Selection Sort: \n");
    printArray(arr, n);
    return 0;
}
```

Penjelasan

1. Memulai dengan indeks pertama dan menemukan elemen terkecil di seluruh array.
2. Menukar elemen terkecil dengan elemen pada indeks pertama.
3. Melanjutkan ke elemen berikutnya dan mengulangi proses sampai seluruh array terurut.

2. Merge Sort

Merge Sort bekerja dengan prinsip divide and conquer. Array dibagi menjadi dua bagian, kemudian masing-masing bagian diurutkan secara rekursif, dan akhirnya kedua bagian yang sudah terurut digabungkan kembali menjadi satu array yang terurut.

Source Code

```
//created by Fhandy Nofalino Akhsan 23343065

#include <stdio.h>

// Fungsi untuk menggabungkan dua subarray dari arr[]
void merge(int arr[], int l, int m, int r) {
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;

    // Membuat array sementara
    int L[n1], R[n2];

    // Menyalin data ke array sementara L[] dan R[]
    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    // Menggabungkan kembali array sementara ke arr[l..r]
    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++;
    }

    // Menyalin sisa elemen L[], jika ada
    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }

    // Menyalin sisa elemen R[], jika ada
    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

// Fungsi utama untuk mengimplementasikan Merge Sort
void mergeSort(int arr[], int l, int r) {
```

```
    if (l < r) {
        // Menemukan titik tengah
        int m = l + (r - l) / 2;

        // Mengurutkan bagian pertama dan kedua
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);

        // Menggabungkan bagian yang terurut
        merge(arr, l, m, r);
    }
}

// Fungsi untuk mencetak array
void printArray(int A[], int size) {
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", A[i]);
    printf("\n");
}

// Fungsi utama
int main() {
    int arr[] = {12, 11, 13, 5, 6, 7};
    int arr_size = sizeof(arr) / sizeof(arr[0]);

    printf("Array sebelum disorting: \n");
    printArray(arr, arr_size);

    mergeSort(arr, 0, arr_size - 1);
```

```
    printf("Array setelah disorting dengan Merge Sort: \n");
    printArray(arr, arr_size);
    return 0;
}
```

Penejelsan

1. Membagi array menjadi dua bagian secara rekursif sampai setiap bagian hanya memiliki satu elemen.

2. Menggabungkan dua bagian yang sudah diurutkan dengan cara membandingkan elemen dari kedua bagian dan menyusunnya kembali ke dalam array utama.

3. Proses ini berlanjut sampai seluruh array digabungkan kembali dan terurut.

Kesimpulan

Kedua algoritma memiliki karakteristik dan keunggulan masing-masing. Selection Sort lebih sederhana namun kurang efisien pada array yang besar, sedangkan Merge Sort lebih efisien pada array yang besar namun lebih kompleks dalam implementasinya.

Referensi

1. Sorting Algorithms- Insertion Sort, Selection Sort, Quick Sort, Merge Sort, Bubble Sort | by Pravallika Devireddy | Learning Python programming language | Medium
2. Merge Sort (With Code in Python/C++/Java/C) (programiz.com)
3. [2023] Job Sheet 9 - Selection and Merge Sort.pdf