LAPORAN

“Pendekatan Divide and Conquer

Menggunakan Algoritma Merge Sort”

Di susun Untuk Memenuhi Tugas Mata Kuliah

Desain dan Analisis Algoritma

Dosen Pengampu :

Muhammad Hifdzi Adini, S.Kom., M.T./ Dr. R. Ati Sukmawati, M.Kom.

Kelompok 7 :

Ferzy Triwarsana Putra	2110131310003
Muhammad Rizky Al Farabi	2110131310007
Julita Hasanah	2110131120005
Yuliana	2110131220001

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI

UNIVERSITAS LAMBUNG MANGKURAT

FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN

PENDIDIKAN KOMPUTER

BANJARMASIN

2022

DAFTAR ISI

BAB I

PENDAHULUAN

1.2 Studi Kasus

Divide and Conquer merupakan algoritma yang berprinsip memecah-mecah permasalahan yang terlalu besar menjadi beberapa bagian kecil sehingga lebih mudah untuk diselesaikan. Langkah-langkah umum algoritma Divide and Conquer :

Divide : Membagi masalah menjadi beberapa upa-masalah yang memiliki kemiripan dengan masalah semula namun berukuran lebih kecil (idealnya berukuran hampir sama).

Conquer : Memecahkan (menyelesaikan) masing-masing upa-masalah (secara rekursif).

Combine : Menggabungkan solusi masing-masing upa-masalah sehingga membentuk solusi masalah semula.

Sesuai dengan karakteristik pembagian dan pemecahan masalah, maka algoritma ini dapat berjalan baik pada persoalan yang bertipe rekursif (perulangan dengan memanggil dirinya sendiri). Salah satu penggunaan algoritma ini yang paling populer adalah dalam hal pengolahan data yang bertipe array . Mengapa ? Karena pengolahan array pada umumnya selalu menggunakan prinsip rekursif atau iteratif. Penggunaan secara spesifik adalah untuk mencari nilai minimal dan maksimal serta untuk mengurutkan elemen array.

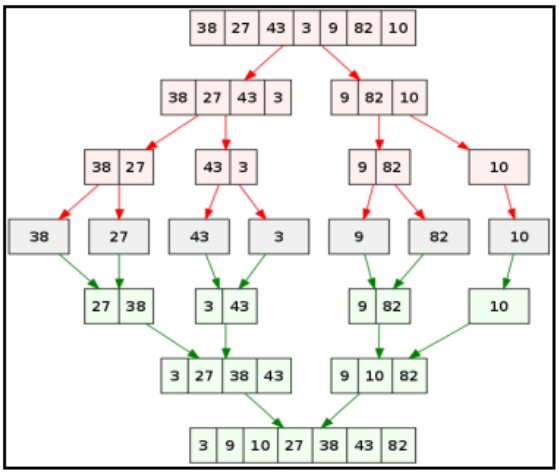
Dalam hal pengurutan ini ada empat macam algoritma pengurutan yang berdasar pada algoritma Divide and Conquer, yaitu merge sort, insert sort, quick sort, dan selection sort. Merge sort dan Quick sort mempunyai kompleksitas algoritma $O(n\ ^{2}\log n)$. Hal ini lebih baik jika dibandingkan dengan pengurutan biasa dengan menggunakan algoritma brute force.

Pada tugas kali ini, kami menggunakan algoritma merge sort untuk mengurutkan dan menghitung berapa banyak bilangan ganjil yang terdapat di dalam sebuah array.

Merge sort merupakan algoritma pengurutan dalam ilmu komputer yang dirancang untuk memenuhi kebutuhan pengurutan atas suatu rangkaian data yang tidak memungkinkan untuk ditampung dalam memori komputer karena jumlahnya yang terlalu besar.

Secara konseptual, sebuah array berukuran n, Merge Sort bekerja sebagai berikut :

1. Jika bernilai 0 atau 1, maka array sudah terurut.
2. Bagi array yang tidak terurut menjadi dua subarray, masing-masing berukuran n/2.
3. Urutkan setiap sub-array. Jika sub-array tidak cukup kecil, lakukan rekursif langkah 2 terhadap sub-array.
4. Menggabungkan dua sub-array kembali menjadi satu array yang terurut.



Contoh simulasi merge sort

5. Untuk menentukan bilangan ganjil yang ada pada array, maka setelah diurutkan, menyeleksi nilai bilangan ganjil yang ada pada array dan hitung berapa jumlahnya.

BAB II

PEMBAHASAN

2.1 Psudocode

```
function bagi2(arr, l, m, r):
    n1 ← m - l + 1
    n2 ← r - m

    L = [1..n1 + 1] and R = [1..n2 + 1] menjadi array baru

    for i ← 1 to n1
        L[i] ← arr[l + i]

    for j ← 1 to n2
        R[j] ← arr[m + 1 + j]

    i ← 0
    j ← 0
    k ← l

    while i < n1 and j < n2:
        if L[i] ≤ R[j]:
            arr[k] ← L[i]
            i ← i + 1
        else:
            arr[k] = R[j]
            j ← j + 1
        k ← k + 1

    while i < n1:
        arr[k] ← L[i]
        i ← i + 1
        k ← k + 1

    while j < n2:
        arr[k] ← R[j]
        j ← j + 1
        k ← k + 1

function array_bagi2(arr, l, r):
    if l < r:
        m ← l+(r-l)//2
        array_bagi2(arr, l, m)
        array_bagi2(arr, m+1, r)
        bagi2(arr, l, m, r)

arr ← [12, 11, 13, 5, 6, 7]
n ← len(arr)
total ← 0
array_bagi2(arr, 0, n-1)

for i ← 1 to n
    if arr[i]%2 ≠ 0
        total ← total + 1

```

ALGORITMA PSEUDOCODE Mergesort(A[0..n − 1])
//Mengurutkan array A[0..n − 1] dengan mengekrusif menggunakan Mergesort
//Input: Sebuah array A[0..n − 1] dengan beberapa element acak didalamnya
//Output: Array A[0..n − 1] yang sudah disusun berdasarkan urutan yang lebih kecil

```
if n > 1
    copy A[0..n/2 − 1] to B[0..n/2 − 1]
    copy A[n/2..n − 1] to C[0..n/2 − 1]
    Mergesort(B[0..n/2 − 1])
    Mergesort(C[0..n/2 − 1])
    Merge(B, C, A) //untuk Merge ada dibawah algoritmanya
```

ALGORITHM Merge(B[0..p − 1], C[0..q − 1], A[0..p + q − 1])
//Menggabungkan dua array list terurut menjadi satu array list terurut
//Input: Dua buah Array B[0..p − 1] dan C[0..q − 1] keduanya disortir
//Output: Array yang diurutkan A[0..p + q − 1] yang merupakan element dari array B dan array C

```
i ← 0; j ← 0; k ← 0
while i<p and j<q do
    if  B[i] ≤ C[j ]
        A[k]← B[i]; i ← i + 1
    else A[k]← C[j ]; j ← j + 1
    k ← k + 1
    if  i = p
        copy C[j..q − 1] to A[k..p + q − 1]
        else copy B[i..p − 1] to A[k..p + q − 1]
```

ALGORITHM Menentukan Bilangan Ganjil

//Mengidentifikasi element dari sebuah array dengan menggunakan perulangan

//Input: Sebuah Array yang sudah melalui proses Mergesort dan Merge dan merupakan Array acak

//Output: Array yang dibernilai Ganjil

```
A = [……..] //Sebuah array
n ← Banyaknya elemen dari array A
total ← 0
for  i in range (n) :
    if  A[i] %2! = 0 :
            print(A[i])
            total ← total + 1
print(total)
```

2.2 Algoritma

Disini kami mengaplikasikan Algoritmanya menggunakan IDLE dari Python, Seperti berikut:

```
def bagi2(arr, l, m, r):
        n1 = m - l + 1
        n2 = r - m

        L = [0] * (n1)
        R = [0] * (n2)

        for i in range(0, n1):
                L[i] = arr[l + i]

        for j in range(0, n2):
                R[j] = arr[m + 1 + j]

        i = 0
        j = 0
        k = l

        while i < n1 and j < n2:
                if L[i] <= R[j]:
                        arr[k] = L[i]
                        i += 1
                else:
                        arr[k] = R[j]
                        j += 1
                k += 1

        while i < n1:
                arr[k] = L[i]
                i += 1
                k += 1

        while j < n2:
                arr[k] = R[j]
                j += 1
                k += 1
def array_bagi2(arr, l, r):
        if l < r:
                m = l+(r-l)//2
                array_bagi2(arr, l, m)
                array_bagi2(arr, m+1, r)
                bagi2(arr, l, m, r)

arr = [12, 11, 13, 5, 6, 7]
n = len(arr)
total=0
array_bagi2(arr, 0, n-1)
print("Jadi bilangan ganjil dari array adalah")
for i in range(n):
    if arr[i]%2!=0:
        print("%d" % arr[i],end=" ")
        total+=1
print("\ntotal bilangan ganjil dari array adalah",total)
```

2.3 Hasil Output

```
Jadi bilangan ganjil dari array adalah
5 7 11 13
total bilangan ganjil dari array adalah 4
```

2.4 Deskripsi Kode Program

Adapun penjelasan untuk kode program nya yaitu sebagai berikut:

a. Pertama adalah fungsi yang bernama bagi2(nama lainnya Mergesort), dimana pada fungsi ini akan meminta list array dengan panjang keseluruhan elemennya.
b. Di dalamnya terdapat variabel yang bernama n1 dan n2, yang merupakan variabel untuk menyimpan panjang dari elemen pembagian dari sebuah array list.
c. Selanjutnya adalah membuat variabel L dan juga R yang menyimpan perkalian antar list [0] dengan banyaknya jumlah dari elemen yang disimpan pada variabel n1 maupun n2.
d. Selanjutnya terdapat perulangan dengan banyak batasnya adalah banyak dari variabel n1 ataupun n2, dimana didalam perulangan tersebut akan mengubah isi dari list array pada L dan juga R dengan isi array yang pertama diinputkan namun sudah dibagi menjadi dua pada variabel L dan juga R
e. Disini juga terdapat variabel i, j yang diisi dengan nol dan juga k yang diisi dengan panjang dari elemen array list
f. Disini juga kami membuat perulangan dengan menggunakan while sebanyak dari variabel yang ada pada kedua variabelnya dengan membandingkan kedua isi dari listnya dan mengurutkannya
g. Adapun perulangan lainnya untuk menggabungkan kedua isi dari list yang sudah dipecah tadi menjadi satu, dengan mengganti elemennya menggunakan elemen yang sudah digabungkan tadi
h. Kemudian adapula fungsi kedua yang bernama array_bagi2(nama lainnya Merge) yang meminta statement array list dan juga panjang dari elemen list, didalamnya terdapat pengkondisian dimana jika panjang dari elemen list lebih besar dari panjang variabel r maka akan dipanggil fungsi pertama yaitu bagi2 dan disini juga akan memanggil fungsi dirinya sendiri yaitu array_bagi2
i. Untuk kode selanjutnya yaitu terdapat sebuah array list sembarang dan terdapat variabel yang menyimpan Panjang dari array list tersebut
j. Adapula variabel total yang berisi angka nol dan kemudian pemanggilan fungsi dari array_bagi2 dengan element statement dari array yang tadi dan menggunakan variabel yang menyimpan nilai Panjang dari array A

k. Kemudian menampilkan string untuk pemberitahuan pada output nantinya
l. Selanjutnya ada perulangan dengan menggunakan for untuk mengecek apakah bilangan tersebut tidak habis dibagi 2, jika iya maka akan masuk kedalam percabangan if dan nilai total akan ditambah 1 serta elemen yang masuk pada percabangan ini akan ditampilkan dengan perintah print
m. Selanjutnya keluar dari percabangan juga akan ditampilkan nilai dari jumlah akhir dari total

Sedangkan untuk penjelasan outputnya:

a. Akan menampilkan string yang tadi sudah dibuat
b. Akan menampilkan juga nilai dari elemen list yang masuk kedalam percabangan
c. Akan menampilkan string yang juga disertai dengan nilai dari total

BAB III

KESIMPULAN

3.1 Kesimpulan

Algoritma Divide and Conquer dapat membantu untuk memecahkan suatu permasalahan. Konsep dari algoritma ini adalah memecah permasalahan menjadi bagian-bagian kecil, sehingga kasus-kasus atau masalah yang ada dapat dengan mudah untuk diselesaikan.

Berdasarkan tugas yang diberikan di kelas, algoritma Divide dan Conquer dapat juga digunakan untuk menghitung banyaknya jumlah element yang bernilai ganjil, dengan menggunakan mergesort dan juga merge untuk memecah dan menggabungkan Kembali array list untuk mengurutkannya.