

Merge Sort

(Tugas DAA)

Anggota Kelompok:

- | | |
|--------------------------------|------------|
| - Fitriani Jaya | H071231012 |
| - Yusra Erlangga Putra | H071231052 |
| - Andi Fauzan Alwan Pananrangi | H071231091 |

Pseudo code:

ALGORITHM *Mergesort*($A[0..n - 1]$)

//Sorts array $A[0..n - 1]$ by recursive mergesort

//Input: An array $A[0..n - 1]$ of orderable elements

//Output: Array $A[0..n - 1]$ sorted in nondecreasing order

if $n > 1$

 copy $A[0..\lfloor n/2 \rfloor - 1]$ to $B[0..\lfloor n/2 \rfloor - 1]$

 copy $A[\lfloor n/2 \rfloor..n - 1]$ to $C[0..\lceil n/2 \rceil - 1]$

Mergesort($B[0..\lfloor n/2 \rfloor - 1]$)

Mergesort($C[0..\lceil n/2 \rceil - 1]$)

Merge(B, C, A) //see below

ALGORITHM *Merge*($B[0..p - 1], C[0..q - 1], A[0..p + q - 1]$)

//Merges two sorted arrays into one sorted array

//Input: Arrays $B[0..p - 1]$ and $C[0..q - 1]$ both sorted

//Output: Sorted array $A[0..p + q - 1]$ of the elements of B and C

$i \leftarrow 0; j \leftarrow 0; k \leftarrow 0$

while $i < p$ **and** $j < q$ **do**

if $B[i] \leq C[j]$

$A[k] \leftarrow B[i]; i \leftarrow i + 1$

else $A[k] \leftarrow C[j]; j \leftarrow j + 1$

$k \leftarrow k + 1$

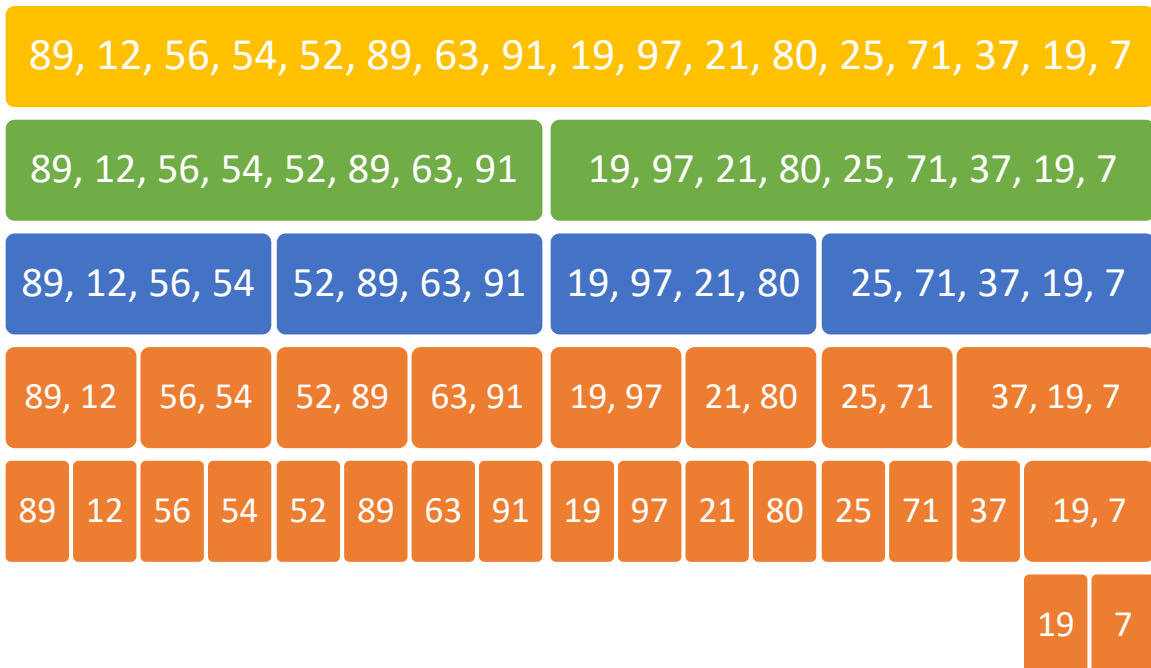
if $i = p$

 copy $C[j..q - 1]$ to $A[k..p + q - 1]$

else copy $B[i..p - 1]$ to $A[k..p + q - 1]$

Penjelasan Cara Kerja :

1. Pembagian data dengan fungsi rekursif dari mergesort()



Tabel-tabel di atas merupakan proses pembagian suatu array atau data hingga tidak dapat dibagi lagi. Proses ini dilakukan oleh program pada method `mergeSort()`. Cara kerjanya ialah method `mergeSort` ini akan membagi dirinya menjadi 2 bagian dengan memisahkan elemen yang paling kiri hingga tengah ($n/2 - 1$) dan elemen tengah ($n/2 - 1$) hingga elemen paling kanan, kemudian memanggil dirinya sendiri terus menerus hingga array-nya sudah tidak dapat dibagi lagi (hingga kondisi `if` tidak terpenuhi).

Setelah pembagian kemudian array yang sudah dibagi tadi dikirim ke dalam fungsi `merge` dalam bentuk address agar array aslinya ikut berubah jika dimanipulasi di dalam fungsi `merge`.

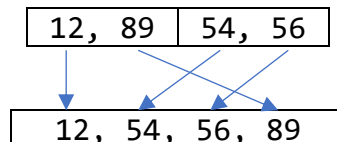
2. Penggabungan data

Pada proses `merge` atau penggabungan sebenarnya terdapat 2 tahap yaitu tahap Penaklukan (`Conquer`) dan Penggabungan (`Merge`).

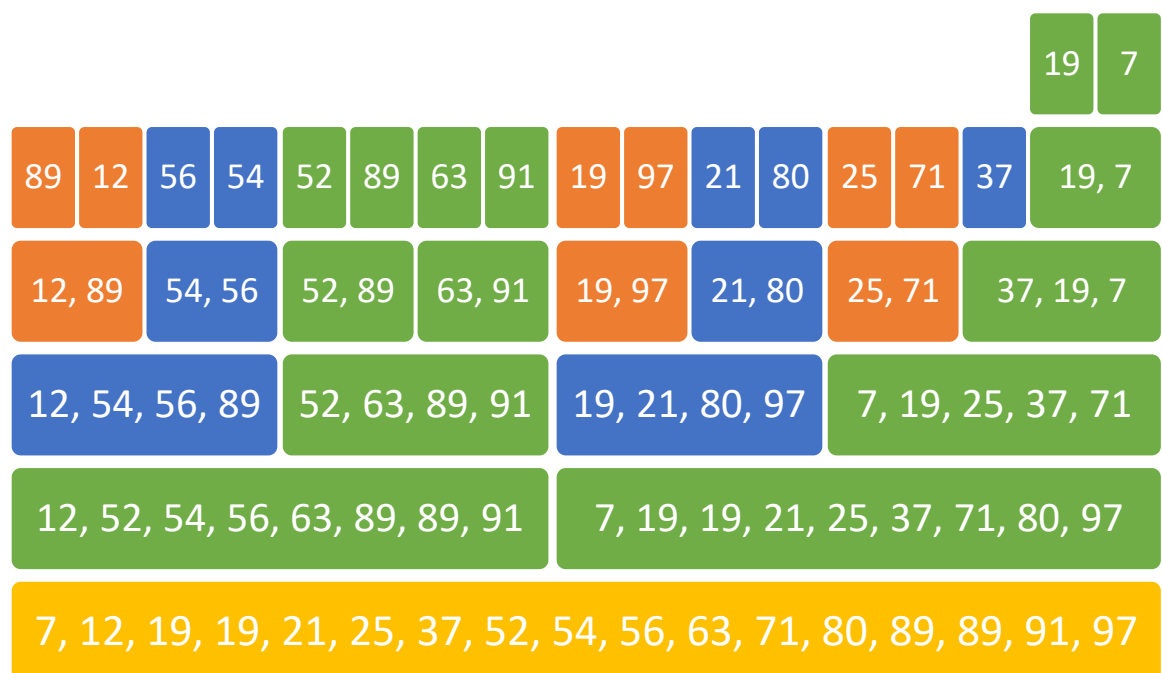
Proses Penaklukan adalah proses mengurutkan setiap elemen yang ada di dalam array yang sudah dibagi tadi, kemudian setelah diurutkan kemudian masuk ke proses penggabungan. Proses penggabungan ialah proses menggabungkan 2 buah array yang telah diurutkan dengan proses penaklukan tadi. Proses ini terus-menerus berulang hingga semua array yang sudah dibagi tadi bergabung menjadi satu array yang terurut.

Secara garis besar, pada proses dalam method merge() akan menerima sebuah parameter array B[left-mid], C[mid-right] dan A[left-right]. Method ini akan menjalankan sebuah looping while yang membandingkan setiap elemen yang dimiliki array B dan array C, elemen terkecil akan dimasukkan satu-persatu ke dalam array A hingga keseluruhan elemennya terurut. Output method ini adalah sebuah array A yang telah terurut yang berasal dari penggabungan array B dan array C tadi.

Contoh salah satu penggabungan dan pengurutan :



Proses penggabungan dan pengurutan sesuai urutan program berjalan adalah sebagai berikut :



- merge [89 12]
sort -> 12 89
- merge [56 54]
sort -> 54 56
- merge [89 12 56 54]
sort -> 12 54 56 89
- merge [52 89]
sort -> 52 89
- merge [63 91]
sort -> 63 91

- merge [52 89 63 91]
sort -> 52 63 89 91
- merge [89 12 56 54 52 89 63 91]
sort -> 12 52 54 56 63 89 89 91
- merge [19 97]
sort -> 19 97
- merge [21 80]
sort -> 21 80
- merge [19 97 21 80]
sort -> 19 21 80 97
- merge [25 71]
sort -> 25 71
- merge [19 7]
sort -> 7 19
- merge [37 19 7]
sort -> 7 19 37
- merge [25 71 37 19 7]
sort -> 7 19 25 37 71
- merge [19 97 21 80 25 71 37 19 7]
sort -> 7 19 19 21 25 37 71 80 97
- merge [89 12 56 54 52 89 63 91 19 97 21 80 25 71 37 19 7]
sort -> 7 12 19 19 21 25 37 52 54 56 63 71 80 89 89 91 97

Berikut adalah contoh diagram secara menyeluruh tanpa berpatokan pada urutan iterasi program :

