**Lembar Kerja Topik Merge Sorting**

Topik : Merge shorting

Anggota Majelis :

|  |  |  |  |
| --- | --- | --- | --- |
| No | NIM | Nama | Asal Kelompok |
| 1 | 2106594 | Aditya Mulyana | 1 |
| 2 | 2103670 | Galih Lazuardi | 4 |
| 3 | 2013592 | M. Raihan B. | 2 |
| 4 | 2100016 | Wendi Kardian | 3 |

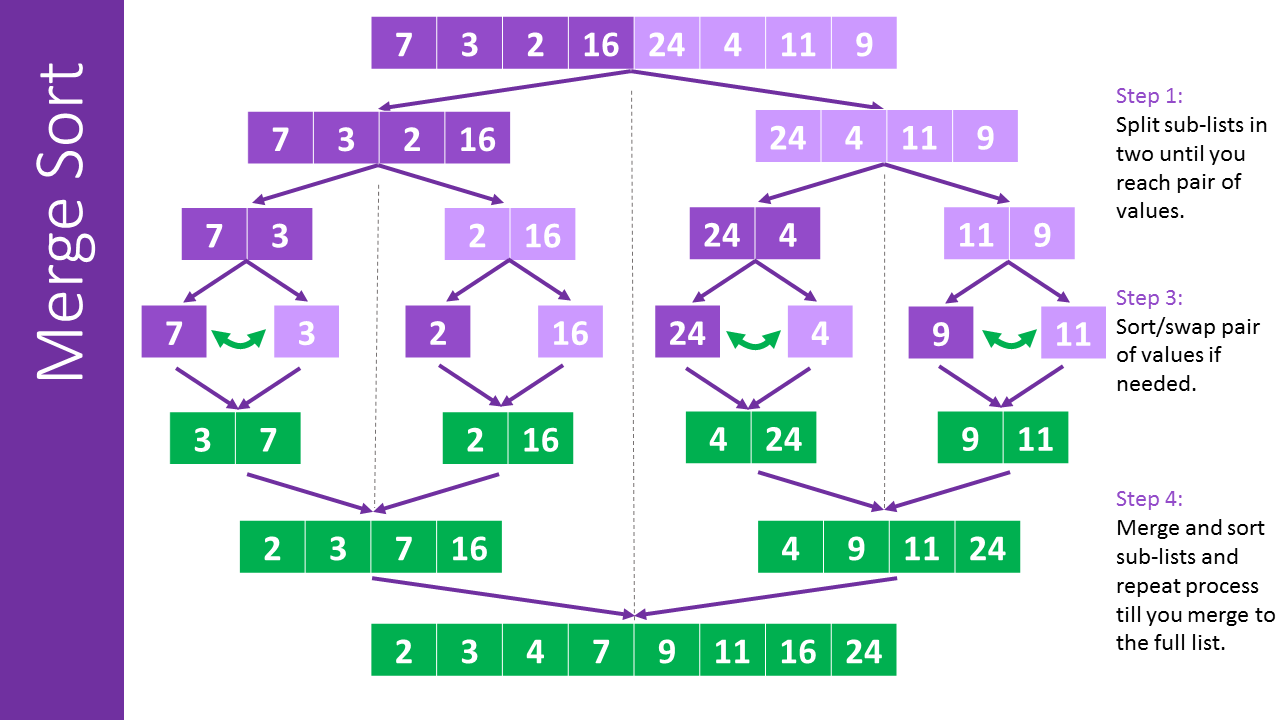
1. Jelaskan secara singkat tujuan dari algoritma yang Anda pilih

Algoritma merge sorting merupakan sebuah algoritma yang umum digunakan untuk mengurutkan sejumlah data, dengan metode memecah data menjadi bagian terkecil lalu disatukan kembali dengan aturan data harus teratur baik ascending maupun descending.

1. Jelaskan tahapan/cara kerja Algoritma yang Anda pilih

* Pecah deret data menjadi dua bagian dan ulangi terus menerus sampai mendapatkan bagian terkecil dari data.
* Kemudian tukar kan data sesuai urutan yang diinginkan, baik ascending maupun descending,.
* Lalu gabungkan kembali data sambil mengurutkan kembali data. Lalu ulangi proses tersebut sampai data terurut dan tergabung sepenuhnya.

1. Ilustrasikan (bisa berupa gambar) bagaimana proses algoritma tersebut! (Penjelasan ilustrasi akan menjadi bahan yang akan Anda sajikan ke anggota kelompok Asal. Penjelasan Ilustasi dapat disajikan dengan kreatif menggunakan media apapun (misal: animasi dalam power point).



Untuk penjelasan lebih mendalam ada di lampiran : Ilustrasi Merge Sort.pdf

1. Tuliskan pseudocode algoritma Anda

|  |  |
| --- | --- |
| Inisialisasi | /\* C program for Merge Sort \*/  #include <stdio.h>  #include <stdlib.h>  // Merges two subarrays of arr[].  // First subarray is arr[l..m]  // Second subarray is arr[m+1..r]  void merge(int arr[], int l, int m, int r)  {  int i, j, k;  int n1 = m - l + 1;  int n2 = r - m;  /\* create temp arrays \*/  int L[n1], R[n2];  /\* Copy data to temp arrays L[] and R[] \*/  for (i = 0; i < n1; i++)  L[i] = arr[l + i];  for (j = 0; j < n2; j++)  R[j] = arr[m + 1 + j];  /\* Merge the temp arrays back into arr[l..r]\*/  i = 0; // Initial index of first subarray  j = 0; // Initial index of second subarray  k = l; // Initial index of merged subarray  while (i < n1 && j < n2) {  if (L[i] <= R[j]) {  arr[k] = L[i];  i++;  }  else {  arr[k] = R[j];  j++;  }  k++;  }  /\* Copy the remaining elements of L[], if there  are any \*/  while (i < n1) {  arr[k] = L[i];  i++;  k++;  }  /\* Copy the remaining elements of R[], if there  are any \*/  while (j < n2) {  arr[k] = R[j];  j++;  k++;  }  }  /\* l is for left index and r is right index of the  sub-array of arr to be sorted \*/  void mergeSort(int arr[], int l, int r)  {  if (l < r) {  // Same as (l+r)/2, but avoids overflow for  // large l and h  int m = l + (r - l) / 2;  // Sort first and second halves  mergeSort(arr, l, m);  mergeSort(arr, m + 1, r);  merge(arr, l, m, r);  }  }  /\* UTILITY FUNCTIONS \*/  /\* Function to print an array \*/  void printArray(int A[], int size)  {  int i;  for (i = 0; i < size; i++)  printf("%d ", A[i]);  printf("\n");  }  /\* Driver code \*/  int main()  {  int arr[] = { 12, 11, 13, 5, 6, 7 };  int arr\_size = sizeof(arr) / sizeof(arr[0]);  printf("Given array is \n");  printArray(arr, arr\_size);  mergeSort(arr, 0, arr\_size - 1);  printf("\nSorted array is \n");  printArray(arr, arr\_size);  return 0;  } |
| Proses | MergeSort(arr[], l, r)  If r > l  1. Find the middle point to divide the array into two halves:  middle m = l+ (r-l)/2  2. Call mergeSort for first half:  Call mergeSort(arr, l, m)  3. Call mergeSort for second half:  Call mergeSort(arr, m+1, r)  4. Merge the two halves sorted in step 2 and 3:  Call merge(arr, l, m, r)  Cara kerja dari merge sort adalah membagi list menjadi sub-sublist menggunakan algoritma recursion, jadi mergesort menerima 3 buah parameter pada umumnya diantaranya adalah array yang ingin dibagi menjadi sub-array, index awal, kemudian index akhir. Proses recursion dalam merge sort ini mengubah posisi index awal dan akhir dari array tersebut. Kemudian, Ketika sudah menjadi subarray yang paling kecil lalu akan dimerge data akan menjadi terurut. |
| Output |  |

1. Pastikan setiap anggota dalam majelis menguasai materi tersebut karena akan menjelaskan disetiap kelompok masing-masing.

**Evaluasi Majelis**

**Tanggal Evaluasi :**

Evaluasi Majelis dilakukan setelah setiap anggota majelis menjelaskan materi yang dibahas ke kelompok masing-masing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama | Asal Kelompok | Evaluasi: | **Tindak lanjut**  (Diisi setelah hasil diskusi dievaluasi kembali dalam majelis) |
|  |  |  | * Berapa anggota kelompok yang sudah paham materi yang disampaikan: * Bagian yang menjadi kendala ketika materi disampaikan: |  |
|  |  |  |  |  |
|  |  |  |  |  |