Sorting

Sorting is huge in the coding world because it keeps things organized and easier to read. If you need names in alphabetical order or need to look at how much money your company has/spent, sorting them is the easiest way to do it. Two of the algorithms that go along with sorting are merge and insertion. Merge sort is a very flexible kind of sort, and it is like divide and conquer. What is basically does is it breaks an array down into smaller pieces and them sorts those pieces. Below is the merge code along with a merge diagram.

{

void merge(int arr[], int lt, int mid, int rt)

{

int low = mid - lt + 1;

int high = rt- mid;

int L[] = new int[low];

int R[] = new int[high];

int i = 0, x = 0;

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

{

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

}

for (x = 0; x < high; x++)

{

R[x] = arr[mid + 1 + x];

}

int k = lt;

i = 0;

x = 0;

while (i < low && x < high)

{

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

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[x];

x++;

}

k++;

}

while (i < low)

{

arr[k] = L[i];

i++;

k++;

}

while (x < high)

{

arr[k] = R[x];

x++;

k++;

}

}

void mergeSort(int arr[], int lt, int rt)

{

int mid;

if (lt < rt) {

mid = (lt + rt) / 2;

mergeSort(arr, lt, mid);

mergeSort(arr, mid + 1, rt);

merge(arr, lt, mid, rt);

}

}

void display(int arr[])

{

for (int i=0; i<arr.length; ++i)

{

System.out.print(arr[i]+" ");

}

}

public static void main(String args[])

{

int arr[] = { 9, 3, 1, 5, 13, 12 };

Sort ob = new Sort();

ob.mergeSort(arr, 0, arr.length - 1);

ob.display(arr);

}

}

Merge Diagram

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 9 | 4 | 3 | 8 | 5 | 1 |

1 3 4 5 8 9

1 5 8

3 4 9

1

5 8

3

4 9

3

1

8 5

9 4

9 4 3

1 8 5

The other method was the insertion method and this one is very simple compared to the merge, what this one does is it picks an array to be a key and then compares it to the other arrays and sees if it is smaller than the key or not and then repeats this process until it is all sorted.

{

static void insertionSort(int arr[], int n)

{

if (n <= 1)

{

return;

}

insertionSort( arr, n-1 );

int last = arr[n-1];

int x = n-2;

while (x >= 0 && arr[x] > last)

{

arr[x+1] = arr[x];

x--;

}

arr[x+1] = last;

}

void display(int arr[])

{

for (int i=0; i<arr.length; ++i)

{

System.out.print(arr[i]+" ");

}

}

public static void main(String[] args)

{

int arr[] = {222, 12, 110, 105, 26};

insertionSort(arr, arr.length);

Sort ob = new Sort();

ob.display(arr);

}

}