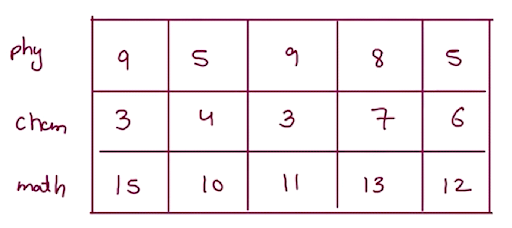
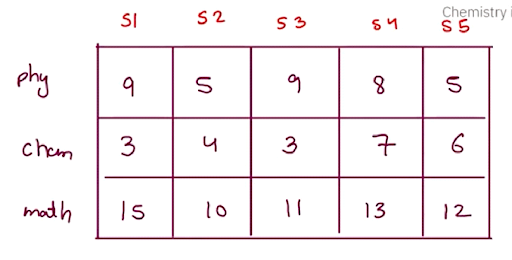
**1. Problem Discussion:**

We have given marks of n students in Physics, Chemistry and Maths. Now, we have 3 operations do to on it these are: First -) We have to sort the students in ascending order of their Physics marks. Second -) After the first operation, we have to sort the students having the same marks in physics in the descending order of their Chemistry marks. Third -) After the second operation, now we have to sort the students having the same marks in Physics and Chemistry in ascending order of their Maths marks. We have to complete the function customSort() which takes arguments phy[], chem[], math[]. The function is for sorting the marks in described order and the final changes should be made in given arrays only.

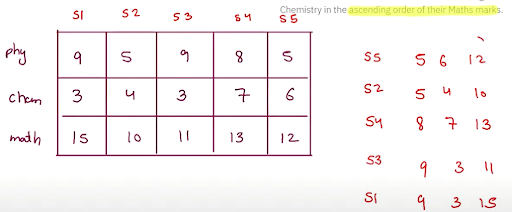
****

Approach:

So, we have given n students marks in three arrays one is for physics marks, second is for chemistry marks and the third is for maths marks. And we have to sort the n students based on their marks in physics, chemistry and maths.

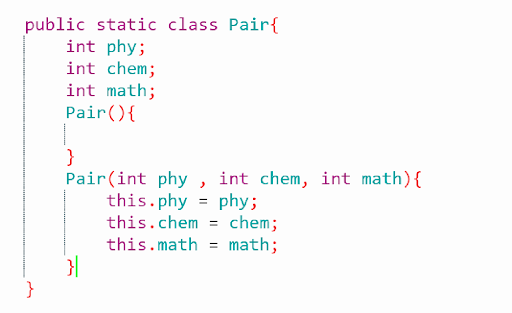
****

Now we have sorting order in which first they will be sort in ascending order of their physics marks i.e. the student having lowest marks in physics will come first. If the marks are equal in physics then the second thing is to sort the students according to their chemistry marks in descending order i.e. the student having highest marks in chemistry will come first. Now, if the marks are equal in chemistry also then we need to sort the students according to the maths marks in ascending order i.e. the student having lowest marks in maths will come first.

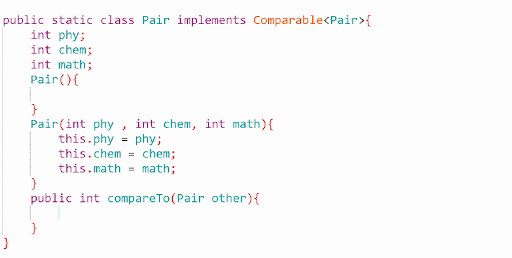
****

These are the three things we have to do and in all of them we just need to sort the arrays where each student has three things i.e. physics marks, chemistry marks and maths marks.

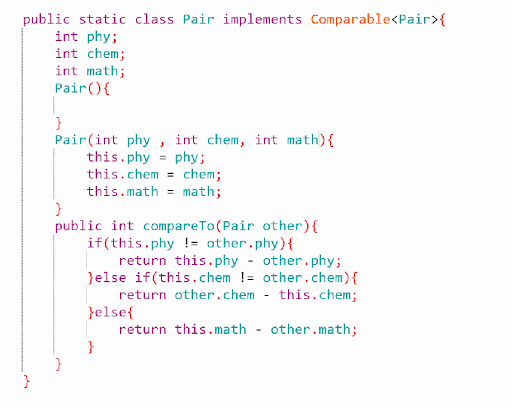
● We will make a Pair class which will have the three things i.e. physics marks, chemistry marks and maths marks to store all the marks of a student in one class.

****

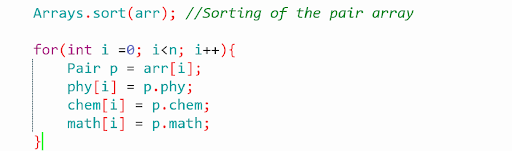
● By making a Pair class we haved wrap up the all marks of a student in one object. Now, we will make an array of Pair class which will store all the marks of three subjects of each student in one object so that we can easily sort it. ● Now we have Pair class array but the array is of Pair class and the inbuilt sort functions doesn't know how to sort two Pair class of students marks so, we have to make the Pair class comparable type so that the sort functions works properly as we want.

****

● In the comparable class we will compare the marks according to the three operation we have given. There will two student’s marks one is this student and another is other student’s marks. Now we will compare the class by writing compareTo(Pair other) function in the Pair class. The comparison of the marks will be : The first operation is to sort them according to physics marks in ascending order. if(this.phy != other.phy){ return this.phy- other.phy; } In the above condition if the physics marks of this and other are same than it will go for the second operation i.e. sort them according to the chemistry marks in descending order. else if(this.chem != other.chem){ return other.chem - this.chem; } In the above condition the chemistry marks are also same than the pair class array will be compared according to the third operation i.e. sort them according to the maths marks. else{ return this.math - other.math; }

****

● Now we have sorted the pair class and we have the sorted students but the second point is we have to make changes in the given arrays only. Now for this we again traverse in the Pair array and at ith position we will update the marks of phy[i] , chem[i] and math[i] arrays.

****

Code:

ConsoleJava

import java.util.\*;

import java.io.\*;

public class Main {

public static class Marks implements Comparable<Marks>{

int phy;

int chem;

int math;

Marks(){

}

Marks(int phy, int chem, int math){

this.phy=phy;

this.chem=chem;

this.math=math;

}

public int compareTo(Marks o){

if(this.phy!=o.phy){

return this.phy-o.phy;

}else if(this.chem!=o.chem){

return o.chem-this.chem;

}else{

return this.math-o.math;

}

}

}

/\*You have to complete the body of customSort function,

after sorting final changes should be made in given arrays only. \*/

public static void customSort(int[]phy,int[]chem,int[]math) {

//write your code here

int N=phy.length;

Marks arr[]=new Marks[N];

//fill

for(int i=0; i<N; ++i){

arr[i]=new Marks(phy[i], chem[i], math[i]);

}

Arrays.sort(arr);

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

phy[i]=arr[i].phy;

chem[i]=arr[i].chem;

math[i]=arr[i].math;

}

}

public static void main(String[]args) {

Scanner scn = new Scanner(System.in);

//input work

int N = scn.nextInt();

int[]phy = new int[N];

int[]chem = new int[N];

int[]math = new int[N];

for(int i=0; i < N;i++) {

phy[i] = scn.nextInt();

}

for(int i=0; i < N;i++) {

chem[i] = scn.nextInt();

}

for(int i=0; i < N;i++) {

math[i] = scn.nextInt();

}

customSort(phy,chem,math);

//output

for(int i=0; i < N;i++) {

System.out.println(phy[i] + " " + chem[i] + " " + math[i]);

}

}

}

Analysis:

Time Complexity: O(nlogn), for sorting the array we have used sort() function which takes O(nlog(n)) time.

Space Complexity: O(n)