**Sorting** – arranging data in increasing or decreasing order

Inbuilt method in java for sorting

Arrays.sort(pass the respective array here) 🡪 return type is void, it make changes in original array.

Internally it uses two sorting algorithm

1.Primitive types(int , float ,char)

Algorithm 🡪 Dual-pivot Quicksort (optimised version of quicksort (it takes two pivot elements) )

Worst-Case T.C 🡪 O(n^2) but for most cases it will be O(nlogn) but bad input can cause O(n^2)

Space complexity 🡪 O(logn)

2.Object Types (String[] , Integer[] )

Algorithm 🡪 Timsort (made using merge sort and insertion sort)

Worst-Case T.C 🡪 O(nlogn)

Space complexity 🡪 O(n)

order of removal

Q- N elements in array, you have to remove all array elements in a way so that cost will be minimum. Cost of removing per element = sum of all elements present in array

1. Remove the maximum element first, otherwise it will be added in cost every time which will increase the cost.

Sort the array in increasing order then start removing from last index

Good integer

Q- n elements in array. Find no. of good integer present in that Array. A element will become a good integer, if count of smaller elements than that element is equal to that element .

1. If elements are not repeated in the array

Steps🡪

Sort the array

Match the number with the index (if matched good integer, if not matched then not a good integer)

1. If Elements will repeat

Sort the array

Match the first occurrence of every element with index(if element is good Integer, rest repeated element will also be a good Integer)

**RECURSION**

How to make recursive condition

1. Faith- That this function will do its work…
2. So if this function will do its work, then break your problem into sub-problem and call that

function.(it will do all the calculation automatically till that iteration)…. If I get answer of all

previous sub-problems.

1. Calculate answer for your current problem. (calculate answer for current iteration only).

Consider answer of both(2nd and 3rd step) according to the situation and return your answer.

**STRING**

How to take character input as array

char c = sc.nextLine().charAt(index) ;

TYPE CONVERSION

NOTE- if you are directly storing values compiler will do automatic type-conversion after performing operation(+,-,/,\*) on values , compiler will not do typecasting, we have to do explicit type-conversion.

char to int – implicit type conversion happen always, because

before operation on values, compiler will do implicit type conversion always

but after performing operation on values like

char+int int value will come

int+int int value will come

and you are storing the value in int variable only.

So no need of explicit typecasting in any case .

int to char – not always implicit type-conversion will happen because

before operation on values compiler will do type-conversion implicitly but

after performing operation on values

char + int int value will come

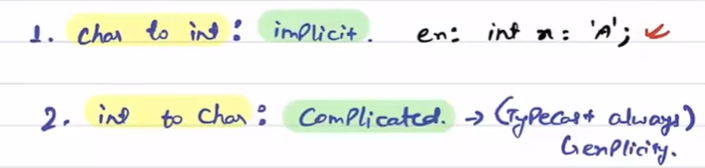
int + int int value will come

and you are storing int value to char variable(complier will not do type-conversion), so

we have to do explicit typecasting.

And in case of copying int type variable to char variable, explicit typecasting is needed.

Solution – in “int to char” case always do explicit typecasting.



Avoid adding characters in String …why? Because every time we add a character in String… it is not added in current String…. a new string is created in another memory, all the characters of old String is copy into that new String and new character is also added in that new String… this process is repeated every time we add a character in String

So instead convert String into char array and then do further changes in that array.

String is immutable

Length of String-

int l = s.length() ; returns length of the string

Accessing particular element of string-

char c = s.charAt(i) ; return character of that particular index

Comparing two string-

boolean b = s1.equals(s2) ; returns true if two String are same else false

String to char array-

Char[] arr = s.toCharArray() ; return char array

Lowercase-

String s1 = s.toLowerCase() ; return String in lowercase

Uppercase-

String s2 = s.toUpperCase() ; return String in uppercase

Substring –

String s1 = s.substring(starting\_index, ending\_index) ;

Comparing two String but ignore case-

boolean b = s.equalsIgnoreCase(s2) ; return true if both String are same else false

This will split the string into string of array

String[] s2 = s.split(by which character you want to split) returns String array

This will convert string into int

int a = Integer.parseInt(String) ;

Char array to string

String ans = new String(arr) ;

String ans = s.valueOf(arr) ;

Use any String here, if no String present in program make one String use that String here

In String

== compares the memory address of the string.

equals() is used to compare the content inside strings.

Q- Converting hashset in ArrayList?

ArrayList<Integer> list=new ArrayList<>(hs);

**Pattern**

There are 4 general rules for solving a pattern-based question:

* We always use nested loops for printing the patterns. For the outer loop, we count the number of lines/rows and loop for them.
* Next, for the inner loop, we focus on the number of columns and somehow connect them to the rows by forming a logic such that for each row we get the required number of columns to be printed.
* We print the ‘\*’ inside the inner loop.
* In case there are 2 pattern mixed , then Print 1 pattern, then second pattern.

To make formula use 🡪 use n , i , or constant(like 1,2 etc)

Row star

1. 1
2. 3
3. 5

If number are increasing by two ,Formula will be 🡪 2\*i-1 not always but modify this formula only and you will get answer.

Row star

1 5

2 3

3 1

If number are decresing by two ,Formula will be 🡪 ( 2\*n) - ( 2\*i) + 1 not always but modify this only and you will get answer.

For revision see question no🡪 7,8,9,10,19,20,21(from striver sheet AtoZ)

Methods

Math.abs(argument) – which returns the absolute value of a number (i.e., removes the negative sign if the number is negative) ,return type will be the same of argument passed. And work with int,float,double,long