Converting Integer to String –

Of the three ways String.valueOf(i), Integer.toString(i) and “”+i, “”+i is the fastest as well as shortest way.

Converting Character to String –

Character.toString, String.valueOf, ch+””

Converting Character to Integer –

Between Character.getNumericValue(num) and num – ‘0’, num – ‘0’ is the fastest and shortest way.

Math.floor, Math.ceil, Math.pow return double values, not int so while storing results of these in int variable we must have to explicitely convert it to int by using (int).

Math.round return long value for double value argument and int value for float value argument passed.

Math.abs returns int, long, float, double type value for respective type argument passed.

Though an array in Java can be of data type other than int but the index value of array in Java always needs to be an int value.

Arrays.toArray(), Arrays.asList() functions don’t work with primitive data types like int, long.

While working with list when multiple insertion and deletion is required, always use LinkedList and not ArrayList.

Some rarely used important functions –

1. Array.copyOf(arr,n)
2. Arrays.copyOfRange(arr,start,end);
3. Arrays.sort(arr);
4. Arrays.sort(arr,Collections.reverseOrder());

TreeMap deletes Entry with duplicate keys when sorting is based on Key but when sorting is based on values only then it not only deletes entries with duplicate keys but also entries with duplicate values gets removed. In that case if you want to preserve duplicate keys and values, use PriorityQueue<Map.Entry<>>.

For inserting single map entry to PriorityQueue pq, add it like pq.add(new HashMap.SimpleEntry(key, value)).

Though we can’t use primitive data type with Collection classes/interface like List, Stack, Queue, Sets but we can use array of primitive data type (for example List<int> will show error but List<int[]> will not).

For case where n is negative, while changing value of n from negative to positive by multiplying it with -1, do take care of case when n is Integer.MIN\_VALUE as corresponding -1 \* Integer.MIN\_VALUE is out of range for int so we’ll first add 1 to it then do n\*-1 so that it will change to Integer.MAX\_VALUE.

Do remember the range to integer in java -2147483648 to 2147483647 and consider the case when n can be these values where constraint is given -231 <= n <= 231-1.

Use for-each or iterator class while iterating with collection elements.

Get and put operation in HashMap takes O(1) time complexity as it’s implemented using an array in the back and it provides hash keys for accessing the values while in TreeMap it takes O(log n) time complexity for get and put as it uses Tree in the back and it takes O(log n) time to find the position for insertion or deletion.

You can’t modify(add/remove elements) TreeSets while traversing with iterator or for each loop as it causes concurrentModificationException so you can store the elements to be deleted in another TreeSet/HashSet and after traversing outside loop, use removeAll to delete all those elements stored in that another Set.

For checking if a character is digit or not, (ch>=’0’ && ch<=’9’) is faster way than using Character.isDigit(ch).