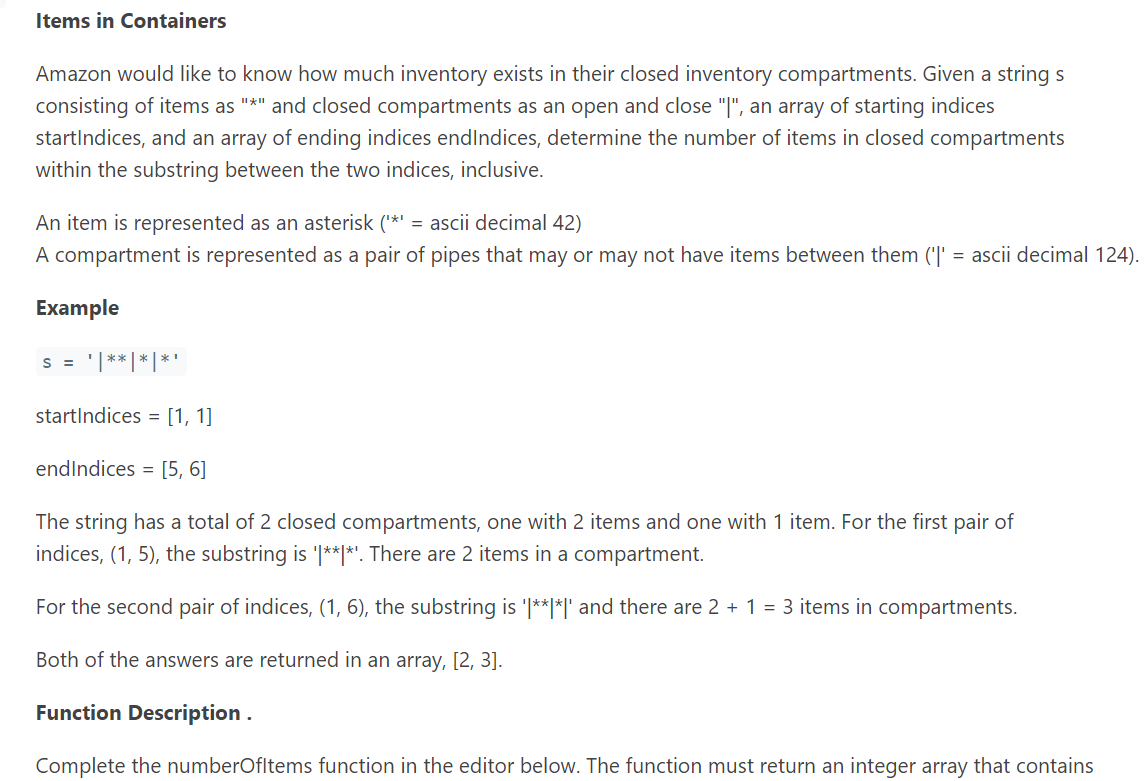
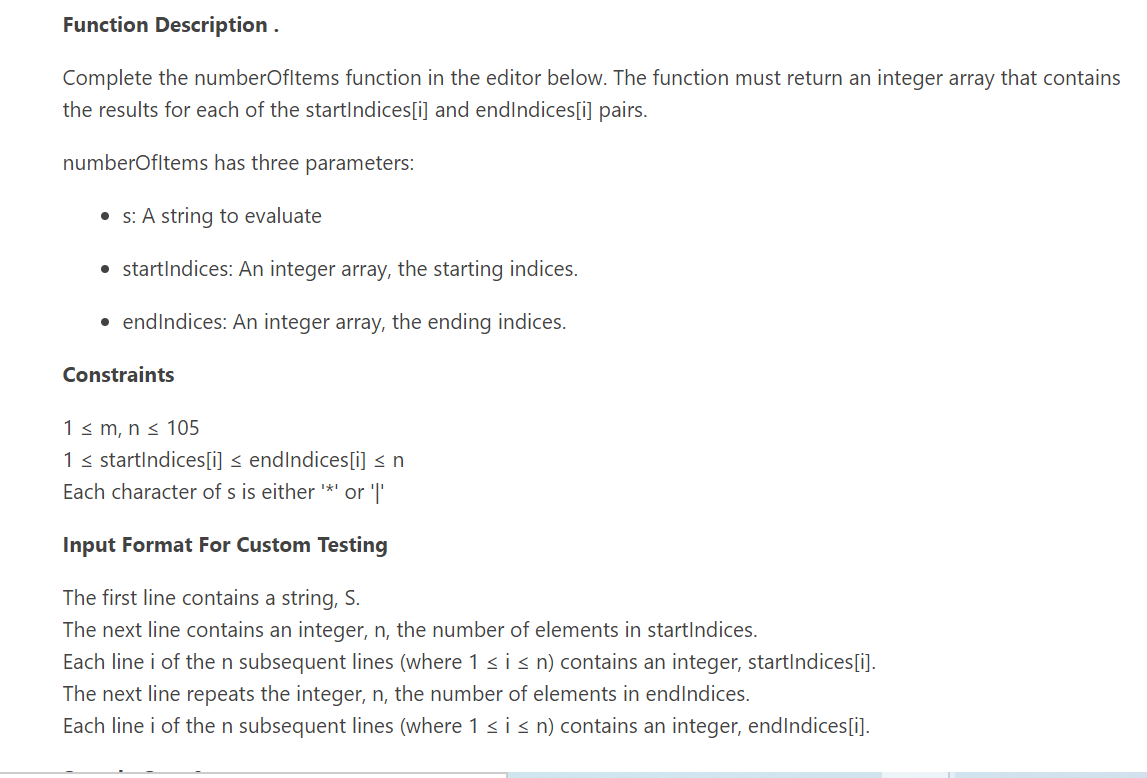
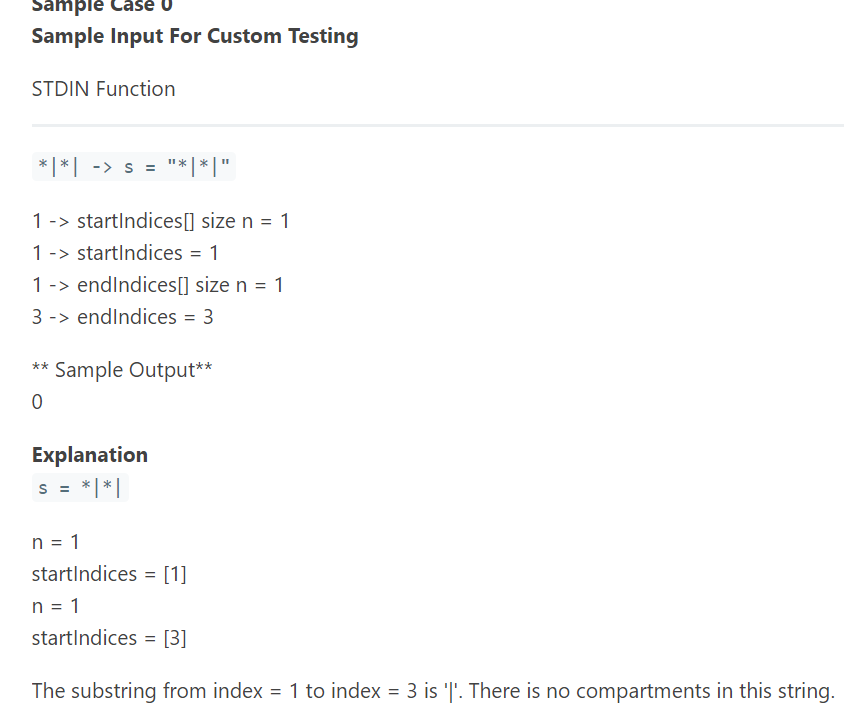
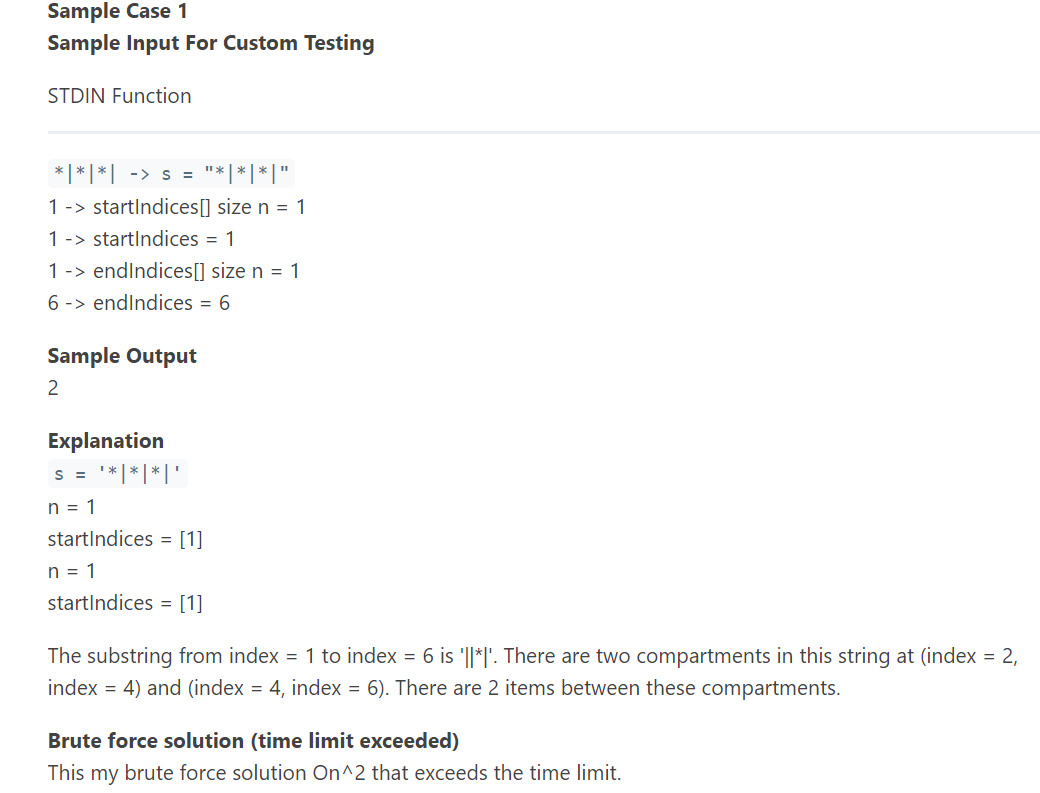
Amazon | OA 2020 | Items in Containers

LEETCODE : <https://leetcode.com/discuss/interview-question/861453/>









Here is the faster approach to the problem using Brute Force. Below is the JAVA solution for the above problem.  
In the worst case, the complexity would be O(n^2).

private static int[] checkItemsInContainer(int[] startIndices, int[] endIndices, String s) {

int[] output = new int[startIndices.length];

int count = 0;

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

count = 0;

String subString = s.substring(startIndices[i] - 1, endIndices[i]);

int initialStartPoint = subString.indexOf('|');

int lastContainer = subString.lastIndexOf('|');

for (int j = initialStartPoint; j < lastContainer; j++) {

char c = s.charAt(j);

if (c == '\*')

count++;

}

output[i] = count;

}

return output;

}

public static List<Integer> numberOfItems(String s, List<Integer> start, List<Integer> end) {

NavigableMap<Integer, Integer> treeMap = new TreeMap<>();

int countSoFar = 0;

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

if (s.charAt(i) == '|') {

treeMap.put(i, countSoFar);

} else {

countSoFar++;

}

}

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

for(int i = 0; i<start.size(); i++) {

list.add(number(treeMap, start.get(i) - 1, end.get(i) - 1));

}

return list;

}

static int number(NavigableMap<Integer, Integer> treemap, int start, int end) {

if (treemap.floorEntry(end) == null || treemap.ceilingEntry(start) == null)

return 0;

int i = treemap.floorEntry(end).getValue() - treemap.ceilingEntry(start).getValue();

return Math.max(i, 0);

}