

ABHILASH.SATISH.WADEKAR.

1RV24MC002 (USN)

RVCE24MCA106 (SAP)

MCA Ist sem A - division

1. Best Time to Buy & Sell Stock.

problem statement : Given array prices[] where prices[i] represents the stock price on day i,

- Find maximum profit by choosing 1 day to buy & 1 day to sell.
- Must buy before you sell.

Input : prices = [7, 1, 5, 3, 6, 4].

Day 0 prices = 7

Day 1 prices = 1.

Day 2 prices = 5

Day 3 prices = 3

Day 4 prices = 6

Day 5 prices = 4

★ Buy on Day 1 (price = 1)

★ Sell on Day 4 (price = 6)

★ profit = 6 - 1 = 5

Iteration Representation:-

for prices = [7, 1, 5, 3, 6, 4]

Day	Price	minPrice (sofar)	profit today (cost)	maxProfit (sofar)
0	7	7	0	0
1	1	1	0	0
2	5	1	5-1=4	4
3	3	1	3-1=2	4
4	6	1	6-1=5	5
5	4	1	4-1=3	5

Java code:

```

class BestTimeToBuyAndSell {
    public static int maxProfit(int[] prices) {
        int minPrice = prices[0];
        int maxProfit = 0;

        for (int i = 1; i < prices.length; i++) {
            int cost = prices[i] - minPrice;
            maxProfit = Math.max(
                maxProfit, cost);
        }
    }
}

```

minPrice = Math.min(minPrice, prices[i]);

return maxProfit;

public static void main (String[] args) {

int[] prices = {7, 1, 5, 3, 6, 4};

System.out.println ("Maximum Profit: "

+ maxProfit (prices));

}

* Time Complexity : $O(n)$

* Space Complexity : $O(1)$.

2. Check if two given strings are isomorphic to each other.

• What is Isomorphism in Strings?

- ▷ Two strings s & t are called isomorphic
- Each character in s can be replaced to get t
- No two characters in s map to the same character in t
- The order of characters must be preserved.

Example Cases:-

Input Strings	Isomorphic?	Explanation
egg, add	Yes	$e \rightarrow a, g \rightarrow d$ (one to one mapping)
foo, bar	No	$o \rightarrow a$ & $o \rightarrow r$ (conflict in mapping)
paper, title	Yes	$p \rightarrow t, a \rightarrow i,$ $e \rightarrow l$

Approach:

- ▷ Use two arrays to store character mappings
 - mapST for $s \rightarrow t$ mapping
 - mapTS for $t \rightarrow s$ mapping
 - ▷ Initialise both arrays with -1, indicating no mapping
 - ▷ Iterate through each character in the strings
 - Check if a mapping exists
 - If mapping is consistent, return true
 - If no mapping exists, create one
- Final Algorithm:
1. Check if lengths of strings s & t are different. If yes, return false.
 2. Create two arrays of size 256 & initialise to -1.
 3. Loop through each character in s & t simultaneously.
 4. For each character pair
 - If the current character already has a mapping, check if it's consistent.

• If it is consistent, return false

• If no mapping exists, create a new mapping

5. If the loop completes, return true

Complexity Analysis:

Time Complexity: $O(n)$

Space Complexity: $O(1)$

Java code:

```
class IsomorphicString {
    public static boolean isIsomorphic(String s,
                                       String t) {
        int[] mapST = new int(256);
        int[] mapTS = new int(256);

        for (int i = 0; i < s.length(); i++) {
            mapST[i] = -1;
            mapTS[i] = -1;
        }

        for (int i = 0; i < s.length(); i++) {
            char chars = s.charAt(i);
            char charT = t.charAt(i);

            if (mapST[chars] != -1 &&
                mapST[chars] != charT)
                return false;
        }
    }
}
```

Best Time

```
if (mapST[charT] != -1 && mapTS[charT] !=
    charS) return false;
```

```
mapST[charS] = charT;
```

```
mapTS[charT] = charS;
```

```
}
```

```
return true;
```

```
}
```

```
public static void main (String[] args) {
```

```
    System.out.println (isIsomorphic("egg", "add"));
```

```
}
```

```
}
```