

Chocolate Distribution Problem

| | |
|----------------------------|---------------|
| ⌵ Difficulty | easy |
| ☑ | ✓ |
| 📅 Finished | @July 8, 2023 |
| ⋮ Problem | array |
| ⋮ Previously asked company | Flipkart |
| ⌵ website | gfg |

Question:

Given an array **A[]** of positive integers of size **N**, where each value represents the number of chocolates in a packet. Each packet can have a variable number of chocolates. There are **M** students, the task is to distribute chocolate packets among **M** students such that :

- 1. Each student gets **exactly** one packet.
- 2. The difference between maximum number of chocolates given to a student and minimum number of chocolates given to a student is minimum.

Example 1:

Input:
N = 8, M = 5
A = {3, 4, 1, 9, 56, 7, 9, 12}
Output:6
Explanation:The minimum difference between maximum chocolates and minimum chocolates is 9 - 3 = 6 by choosing following M packets :{3, 4, 9, 7, 9}.

Example 2:

Input:
N = 7, M = 3
A = {7, 3, 2, 4, 9, 12, 56}
Output:2
Explanation:The minimum difference between maximum chocolates and minimum chocolates is 4 - 2 = 2 by choosing following M packets :{3, 2, 4}.

Optimal solution:

Time complexity: O(nlogn)

Space complexity: O(1)

```
class Solution:
    def findMinDiff(self, A,N,M):
        A.sort()
        n = N-M
        mindiff = A[M-1] - A[0]
        for i in range(n):
            if A[i+M] - A[i+1] < mindiff:
```

```
        mindiff = A[i+M] - A[i+1]  
    return mindiff
```