

5.1 MAXIMUM COINS FROM $3n$ PILES USING GREEDY STRATEGY

Question:

There are $3n$ piles of coins of varying size, you and your friends will take piles of coins as follows: In each step, you will choose any 3 piles of coins (not necessarily consecutive). Of your choice, Alice will pick the pile with the maximum number of coins. You will pick the next pile with the maximum number of coins. Your friend Bob will pick the last pile. Repeat until there are no more piles of coins. Given an array of integers piles where piles[i] is the number of coins in the i th pile. Return the maximum number of coins that you can have.

AIM

To implement a greedy algorithm that maximizes the number of coins you can collect when choosing piles in groups of three.

ALGORITHM

1. Sort the array piles in descending order.
2. In each group of 3 piles:
 - Alice takes the largest
 - You take the second largest
 - Bob takes the smallest
3. Repeat this for n rounds (since there are $3n$ piles).
4. To maximize your coins, sum every second pile in each group of 3 from the sorted array.
 - This corresponds to indices: 1, 3, 5, ..., $2n - 1$

PROGRAM

```
def max_coins(piles):
    piles.sort(reverse=True)
    n = len(piles) // 3
    total = 0
    i = 1
    for _ in range(n):
        total += piles[i]
        i += 2
    return total

piles = list(map(int, input("Enter piles separated by space: ").split()))
print("Maximum coins you can have:", max_coins(piles))
```

Input:

Enter 3n piles separated by space: 2 1 5 3 8 2 9

Output:

```
>>> Enter piles separated by space: 2 1 5 3 8 2 9
Maximum coins you can have: 11
```

RESULT:

Thus the program is successfully executed and the output is verified.

PERFORMANCE ANALYSIS:

- Time Complexity: $O(n \log n)$
- Space Complexity: $O(1)$