CRACK A HACK

MAX SCORE

Course: ALGORITHMIC PROBLEM SOLVING

Course Code: 17ECSE309

Name: Sarjith R M

USN: 01FE15BCS172

PROBLEM STATEMENT

Consider an n-element sequence of integers, $A = \{a_0, a_1, \ldots, a_{n-1}\}$. We want to perform n operations on A, where each operation is defined by the following sequence of steps:

1. Remove any integer, a_i, from A and set it aside.
2. Calculate $score_k = runningSum \bmod a_i$, where $1 \leq k \leq n$ and $runningSum$ is the sum of all the numbers removed from A during the previous $k-1$ operations.
3. Update $runningSum$ such that $runningSum = runningSum + a_i$, where a_i is the integer that was removed from A during step 1 above.

After performing n operations, we sum each $score_k$ to get $totalScore$. In other words:

$$totalScore = \sum_{k=1}^{n} score_k$$

Given n and A, find and print the maximum possible value of $totalScore$ after performing n operations.

Note: The initial values of $runningSum$ and $score_1$ are always 0.

EXPLAINATION WITH EXAMPLE

Sample Input 0

```
3
4 8 5
```

Sample Output 0

```
6
```

Explanation 0

We maximize our score by performing the following $n = 3$ operations:

1. Initially, $runningSum = 0$ and $A = \{4, 8, 5\}$:
   - Remove $a_2 = 5$ from A to get $score_1 = runningSum \bmod a_2 = 0 \bmod 5 = 0$.
   - Add the removed value to $runningSum$ to get $runningSum = 0 + 5 = 5$.
2. $runningSum = 5$ and $A = \{4, 8\}$:
   - Remove $a_1 = 8$ from A to get $score_2 = runningSum \bmod a_1 = 5 \bmod 8 = 5$.
   - Add the removed value to $runningSum$ to get $runningSum = 5 + 8 = 13$.
3. $runningSum = 13$ and $A = \{4\}$:
   - Remove $a_0 = 4$ from A to get $score_3 = runningSum \bmod a_0 = 13 \bmod 4 = 1$.
   - We don't need to update $runningSum$ at this point as we've removed all items from A and cannot perform any more operations.

We then print the result of $score_1 + score_2 + score_3 = 0 + 5 + 1 = 6$ as our answer.

CODE SOLUTION (in PYTHON)

```
#!/bin/python


def getMaxScore(a):
    n = len(a)
    m = 1 << n
    v = [0] * m
    ss = [0] * m
    for l in range(n):
        ml = 1 << l
        al = a[l]
        for i in range(ml, ml + ml):
            it = i ^ ml
            s = al + ss[it]
            rb = v[it] + s % al

            while it:
                itn = it & (it - 1)
```

```
            b = it ^ itn
            it = itn
            r = v[i ^ b] + s % ss[b]
            if r > rb: rb = r
        ss[i] = s
        v[i] = rb

    return v[m - 1]

n = int(raw_input())
a = map(int, raw_input().split())
maxScore = getMaxScore(a)
print(maxScore)
```

REFERENCES

- https://www.hackerrank.com/contests/rookierank-3/challenges/max-score.
- http://codeforces.com/blog/entry/52542
- https://github.com/xploiter-projects/hacker-rank/blob/master/RoockieRank%203/MaxScore.cpp