

AVAILABLE
LESSONS:*Lesson 1*
Iterations*Lesson 2*
Arrays*Lesson 3*
Time Complexity*Lesson 4*
Counting
Elements*Lesson 5*
Prefix Sums*Lesson 6*
Sorting*Lesson 7*
Stacks and
Queues*Lesson 8*
Leader*Lesson 9*
Maximum slice
problem*Lesson 10*
Prime and
composite
numbers*Lesson 11*

RESPECTABLE

NumberSolitaire

START

In a given array, find the subset of maximal sum in which the distance between consecutive elements is at most 6.

Programming language: C++ ▼

A game for one player is played on a board consisting of N consecutive squares, numbered from 0 to $N - 1$. There is a number written on each square. A non-empty zero-indexed array A of N integers contains the numbers written on the squares. Moreover, some squares can be marked during the game.

At the beginning of the game, there is a pebble on square number 0 and this is the only square on the board which is marked. The goal of the game is to move the pebble to square number $N - 1$.

During each turn we throw a six-sided die, with numbers from 1 to 6 on its faces, and consider the number K , which shows on the upper face after the die comes to rest. Then we move the pebble standing on square number I to square number $I + K$, providing that square number $I + K$ exists. If square number $I + K$ does not exist, we throw the die again until we obtain a valid move. Finally, we mark square number $I + K$.

After the game finishes (when the pebble is standing on square number $N - 1$), we calculate the result. The result of the game is the sum of the numbers written on all marked squares.

For example, given the following array:

```
A[0] = 1
A[1] = -2
A[2] = 0
A[3] = 9
A[4] = -1
A[5] = -2
```

one possible game could be as follows:

Sieve of
Eratosthenes

Lesson 12

Euclidean
algorithm

Lesson 13

Fibonacci
numbers

Lesson 14

Binary search
algorithm

Lesson 15

Caterpillar
method

Lesson 16

Greedy
algorithms

Lesson 17

**Dynamic
programming**

Lesson 90

Tasks from
Indeed Prime
2015 challenge

Lesson 91

Tasks from
Indeed Prime
2016 challenge

Lesson 92

Tasks from
Indeed Prime
2016 College
Coders
challenge

Lesson 99

- the pebble is on square number 0, which is marked;
- we throw 3; the pebble moves from square number 0 to square number 3; we mark square number 3;
- we throw 5; the pebble does not move, since there is no square number 8 on the board;
- we throw 2; the pebble moves to square number 5; we mark this square and the game ends.

The marked squares are 0, 3 and 5, so the result of the game is $1 + 9 + (-2) = 8$. This is the maximal possible result that can be achieved on this board.

Write a function:

```
int solution(vector<int> &A);
```

that, given a non-empty zero-indexed array A of N integers, returns the maximal result that can be achieved on the board represented by array A.

For example, given the array

```
A[0] = 1  
A[1] = -2  
A[2] = 0  
A[3] = 9  
A[4] = -1  
A[5] = -2
```

the function should return 8, as explained above.

Assume that:

- N is an integer within the range [2..100,000];
- each element of array A is an integer within the range [-10,000..10,000].

Complexity:

- expected worst-case time complexity is $O(N)$;
- expected worst-case space complexity is $O(N)$, beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Copyright 2009–2016 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

**For
programmers**

Lessons Challenges
Terms FAQ

For companies

Tour About
Pricing us
Blog Jobs
Privacy Terms
Cookies
API

**Sign up for our
newsletter:**

Information about
upcoming challenges,
solutions and lessons
directly in your inbox.

Social:

[f](#) [t](#) [in](#)

Contact us:

For customer support
queries:

UK +44 (0) 208 970
78 68

US 1-415-466-8085
support@codility.com

For sales queries:

UK +44 (0) 208 970
78 67

US 1-415-466-8085
sales@codility.com

© 2009–2016 Codility Ltd., registered in England and Wales (No. 7048726). VAT ID
GB981191408. Registered office: 107 Cheapside, London EC2V 6DN