Minute to Win It

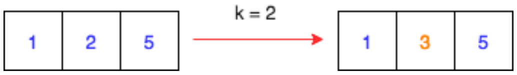
locked

by koca_kodza

Submissions will no longer be placed on the leaderboard. You may still attempt this problem for practice.

In a new version of the game *Minute to Win It*, the math round involves manipulating arrays to meet the given condition. In the challenge, you are given an array of n numbers $[a[0], a[1], \ldots, a[n-1]]$ and an integer k. In one minute, you can change any element of the array to any integer you want. Find the minimum amount of time you have to spend so that the following condition is satisfied: *for all* i *from* 1 *to* $n-1$, $a[i] - a[i-1] = k$.

For example, consider the array $[1, 2, 5]$ and $k = 2$. Then the condition can be satisfied in 1 minute by replacing the 2 with a 3.



Complete the function `minuteToWinIt` which accepts an array a of n integers and an integer k as input and returns the minimum amount of time in minutes.

Input Format

The first line contains two space-separated integers n and k.

The second line contains the array in the form of n space-separated integers $a[0], a[1], \cdots, a[n-1]$.

Constraints

- $2 \leq n \leq 10^5$
- $|k| \leq 10^5$
- $|a[i]| \leq 10^5$

Output Format

Print the minimum number of minutes needed to reorder the array.

Sample Input 0

```
6 2
1 2 5 7 9 85
```

Sample Output 0

```
2
```

Explanation 0

The given array is $[1, 2, 5, 7, 9, 85]$. If we change $2 \Rightarrow 3$ and $85 \Rightarrow 11$ at index 1 and 5 respectively, we get the desired array $[1, 3, 5, 7, 9, 11]$.

Submissions: 5420
Max Score: 11.81
Difficulty: Easy

Rate This Challenge:
☆☆☆☆☆

More

Current Buffer (saved locally, editable)

Erlang

```
-module(solution).
-export([main/0]).
-import(os, [getenv/1]).

% Complete the minuteToWinIt function below.
minuteToWinIt(A, K) ->
    % Return the minimum amount of time in minutes.

read_multiple_lines_as_list_of_strings(N) ->
    read_multiple_lines_as_list_of_strings(N, []).

read_multiple_lines_as_list_of_strings(0, Acc) ->
    lists:reverse(Acc);
read_multiple_lines_as_list_of_strings(N, Acc) when N > 0 ->
    read_multiple_lines_as_list_of_strings(N - 1, [string:chomp(io:get_line("")) | Acc]).

main() ->
    {ok, Fptr} = file:open(getenv("OUTPUT_PATH"), [write]),

    Nk = re:split(string:chomp(io:get_line("")), "\\s+", [{return, list}, trim]),

    {N, _} = string:to_integer(lists:nth(1, Nk)),

    {K, _} = string:to_integer(lists:nth(2, Nk)),

    ATemp = re:split(string:chomp(io:get_line("")), "\\s+", [{return, list}, trim]),

    A = lists:map(fun(X) -> {I, _} = string:to_integer(X), I end, ATemp),

    Result = minuteToWinIt(A, K),

    io:fwrite(Fptr, "~w~n", [Result]),

    file:close(Fptr),

    ok.

```

Line: 1 Col: 1

Upload Code as File   □ Test against custom input   Run Code   Submit Code

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature

PRACTICE COMPETE JOBS LEADERBOARD Search Justinba1010

All Competitions > Week of Code 38 > Minute to Win It

Minute to Win It

locked

by koca_kodza

Problem Submissions Leaderboard Discussions Editorial

Submitted an hour ago • Score: 0.00

Status: Terminated due to timeout

Terminated due to timeout 10s

Download

Test Case #1
Test Case #2
Test Case #4
Test Case #5
Test Case #6
Test Case #7
Test Case #8
Test Case #9
Test Case #10
Test Case #11
Test Case #12
Test Case #13
Test Case #14
Test Case #15
Test Case #16
Test Case #17
Test Case #18
Test Case #19
Test Case #20
Test Case #21
Test Case #22
Test Case #23
Test Case #24
Test Case #25
Test Case #26
Test Case #27
Test Case #28
Test Case #29
Test Case #30

Submitted Code

Language: Python 3

Open in editor

```
#!/bin/python3

import math
import os
import random
import re
import sys

# Complete the minuteToWinIt function below.
def minuteToWinIt(a, k):
    minimum = len(a)
    for i in range(len(a)):
        counter = len(a)-1
        for j in range(len(a)):
            if(i == j): continue
            if a[i] - a[j] == k*(i-j):
                print("%d | %d -> %d == %d" % (i,j, a[i] - a[j],k*(i-j)))
                counter -= 1
            minimum = min(counter, minimum)
    return minimum
if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    nk = input().split()

    n = int(nk[0])

    k = int(nk[1])

    a = list(map(int, input().rstrip().split()))

    result = minuteToWinIt(a, k)

    fptr.write(str(result) + '\n')

    fptr.close()

```

Minute to Win It

locked

by koca_kodza

Problem Submissions Leaderboard Discussions Editorial

Submitted 33 minutes ago • Score: 11.81 Status: Accepted

✓	Test Case #0	✓	Test Case #1	✓	Test Case #2
✓	Test Case #3	✓	Test Case #4	✓	Test Case #5
✓	Test Case #6	✓	Test Case #7	✓	Test Case #8
✓	Test Case #9	✓	Test Case #10	✓	Test Case #11
✓	Test Case #12	✓	Test Case #13	✓	Test Case #14
✓	Test Case #15	✓	Test Case #16	✓	Test Case #17
✓	Test Case #18	✓	Test Case #19	✓	Test Case #20
✓	Test Case #21	✓	Test Case #22	✓	Test Case #23
✓	Test Case #24	✓	Test Case #25	✓	Test Case #26
✓	Test Case #27	✓	Test Case #28	✓	Test Case #29
✓	Test Case #30				

Submitted Code

Language: Python 3 Open in editor

```
#!/bin/python3

import math
import os
import random
import re
import sys

class SetOccurences:
    def __init__(self):
        self.set = set()
        self.occurences = {}
    def add(self, x):
        if x in self.set:
            self.occurences[x] += 1
        else:
            self.occurences[x] = 1
            self.set.add(x)
```

```python
    def getMax(self):
        maximum = 0
        for i in self.occurences:
            maximum = max(self.occurences[i], maximum)
        return maximum

# Complete the minuteToWinIt function below.
def minuteToWinIt(a, k):
    setty = SetOccurences()
    for i in range(len(a)):
        setty.add(a[i] - i*k)
    return len(a) - setty.getMax()


    # Return the minimum amount of time in minutes.

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    nk = input().split()

    n = int(nk[0])

    k = int(nk[1])

    a = list(map(int, input().rstrip().split()))

    result = minuteToWinIt(a, k)

    fptr.write(str(result) + '\n')

    fptr.close()

```

PRACTICE COMPETE JOBS LEADERBOARD Search Justinba1010

All Competitions > Week of Code 38 > Minute to Win It

Minute to Win It

locked

by koca_kodza

Problem | Submissions | Leaderboard | Discussions | Editorial

Submitted a few seconds ago • Score: 11.81

Status: Accepted

✓	Test Case #0	✓	Test Case #1	✓	Test Case #2
✓	Test Case #3	✓	Test Case #4	✓	Test Case #5
✓	Test Case #6	✓	Test Case #7	✓	Test Case #8
✓	Test Case #9	✓	Test Case #10	✓	Test Case #11
✓	Test Case #12	✓	Test Case #13	✓	Test Case #14
✓	Test Case #15	✓	Test Case #16	✓	Test Case #17
✓	Test Case #18	✓	Test Case #19	✓	Test Case #20
✓	Test Case #21	✓	Test Case #22	✓	Test Case #23
✓	Test Case #24	✓	Test Case #25	✓	Test Case #26
✓	Test Case #27	✓	Test Case #28	✓	Test Case #29
✓	Test Case #30				

Submitted Code

Language: Erlang

Open in editor

```erlang
-module(solution).
-export([main/0]).
-import(os, [getenv/1]).

% Complete the minuteToWinIt function below.
minuteToWinIt(A, K) ->
    {Dict, Length} = getDict(A, K, dict:new(), 0),
    Length - getMax(dict:to_list(Dict)).

getMax(List) ->
    [{_,Value}|_] = List,
    getMax(List, Value).

getMax([], Max) -> Max;
getMax([{_, Value}| Tail], Max) when Value > Max ->
    getMax(Tail, Value);
getMax([_| Tail], Max) ->
    getMax(Tail, Max).
```

```erlang


addOne(X) ->
    X + 1.

getDict([], _, Dict, Counter) ->
    {Dict,Counter};
getDict([A|Tail], K, Dict, Counter)->
    case dict:is_key(A - Counter*K, Dict) of
        true -> getDict(Tail, K, dict:update(A-Counter*K, fun (X) -> X+1 end, Dict), Counter+1);
        false -> getDict(Tail, K, dict:store(A-Counter*K, 1, Dict), Counter+1) end.

read_multiple_lines_as_list_of_strings(N) ->
    read_multiple_lines_as_list_of_strings(N, []).

read_multiple_lines_as_list_of_strings(0, Acc) ->
    lists:reverse(Acc);
read_multiple_lines_as_list_of_strings(N, Acc) when N > 0 ->
    read_multiple_lines_as_list_of_strings(N - 1, [string:chomp(io:get_line("")) | Acc]).

main() ->
    {ok, Fptr} = file:open(getenv("OUTPUT_PATH"), [write]),

    Nk = re:split(string:chomp(io:get_line("")), "\\s+", [{return, list}, trim]),

    {N, _} = string:to_integer(lists:nth(1, Nk)),

    {K, _} = string:to_integer(lists:nth(2, Nk)),

    ATemp = re:split(string:chomp(io:get_line("")), "\\s+", [{return, list}, trim]),

    A = lists:map(fun(X) -> {I, _} = string:to_integer(X), I end, ATemp),

    Result = minuteToWinIt(A, K),

    io:fwrite(Fptr, "~w~n", [Result]),

    file:close(Fptr),

    ok.

```

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature