

```
In [17]: 1 import numpy as np
```

LC1904 <https://leetcode.com/problems/the-number-of-full-rounds-you-have-played/>  
[\(https://leetcode.com/problems/the-number-of-full-rounds-you-have-played/\)](https://leetcode.com/problems/the-number-of-full-rounds-you-have-played/)

<https://leetcode.com/discuss/interview-question/398023/Microsoft-Online-Assessment-Questions/884278>  
[\(https://leetcode.com/discuss/interview-question/398023/Microsoft-Online-Assessment-Questions/884278\)](https://leetcode.com/discuss/interview-question/398023/Microsoft-Online-Assessment-Questions/884278)

## Jimmy Tree

<https://stackoverflow.com/questions/62392510/find-and-format-visually-aesthetically-pleasant-pattern-of-trees-in-a-forest-usi> (<https://stackoverflow.com/questions/62392510/find-and-format-visually-aesthetically-pleasant-pattern-of-trees-in-a-forest-usi>) 跟这里基本一致，给一个array代表一排树的高度，然后切一棵树来满足排列是高低相间。问有多少种切法。类似wiggle array。应该是mid难度吧。

The screenshot shows a programming interface with the following details:

- Title:** Task 2
- Time:** 1h 02min
- Language:** Java 8
- Description:** Jimmy owns a garden in which he has planted N trees in a row. After a few years, the trees have grown up and now they have different heights.
- Requirement:** Jimmy pays much attention to the aesthetics of his garden. He finds his trees aesthetically pleasing if they alternately increase and decrease in height (... , shorter, taller, shorter, taller, ...).
- Examples:** These are examples of aesthetically pleasing trees:  
Two bar charts on lined paper. The first chart shows a sequence of heights: short, tall, short, tall. The second chart shows a sequence: tall, short, tall, short, tall, short.
- Non-examples:** These are examples of trees that are not aesthetically pleasing:  
Two bar charts on lined paper. The first chart shows a sequence: short, tall, short, tall, short. The second chart shows a sequence: tall, short, tall, short, tall, short, tall.
- Test Output:** Test 0 | OK, Test 1 | OK, Test 2 | OK, Test 3 | OK, Your example |
- Watermark:** @一亩三分地

wayfair

① Task 2

⌚ 1h 02min

Java 8

Files

task2

soluti...  
test-i...

1 Note that two adjacent trees cannot have equal heights. It may turn out that some trees have to be cut out, in order to keep the remaining trees aesthetically pleasing. However, there is a legal restriction that allows a gardener to cut out at most one tree in his possession. In how many ways can Jimmy cut out exactly one tree, so that the remaining ones are aesthetically pleasing?

Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given an array A consisting of N integers, where A[K] denotes the height of the K-th tree, returns the number of ways of cutting out one tree, so that the remaining trees are aesthetically pleasing. If it is not possible to achieve the desired result, your function should return -1. If the trees are already aesthetically pleasing without any removal, your function should return 0.

Examples:

1. Given A = [3, 4, 5, 3, 7] your function should return 3:

- You can remove A[0] so the sequence becomes [4, 5, 3, 7];
- You can remove A[1] so the sequence becomes [3, 5, 3, 7];
- You can remove A[2] so the sequence becomes [3, 4, 3, 7].

2. Given A = [1, 2, 3, 4] your function should return -1, since there is no single tree that Jimmy can cut out that would leave the rest of the trees looking aesthetically pleasing.

3. Given A = [1, 3, 1, 2] your function should return 0, since the trees are already aesthetically pleasing and no removal is needed.

Assume that:

- N is an integer within the range [4..200];

Autocomplete is connected | All changes saved

Test Output

OK

Example test OK

Example test OK

Your code is example test.

@一亩三分地

wayfair

⌚ 1h 02min

Java 8

Files

task2

soluti...  
test-i...

1 Note that two adjacent trees cannot have equal heights. It may turn out that some trees have to be cut out, in order to keep the remaining trees aesthetically pleasing. However, there is a legal restriction that allows a gardener to cut out at most one tree in his possession. In how many ways can Jimmy cut out exactly one tree, so that the remaining ones are aesthetically pleasing?

Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given an array A consisting of N integers, where A[K] denotes the height of the K-th tree, returns the number of ways of cutting out one tree, so that the remaining trees are aesthetically pleasing. If it is not possible to achieve the desired result, your function should return -1. If the trees are already aesthetically pleasing without any removal, your function should return 0.

Examples:

1. Given A = [3, 4, 5, 3, 7] your function should return 3:

- You can remove A[0] so the sequence becomes [4, 5, 3, 7];
- You can remove A[1] so the sequence becomes [3, 5, 3, 7];
- You can remove A[2] so the sequence becomes [3, 4, 3, 7].

2. Given A = [1, 2, 3, 4] your function should return -1, since there is no single tree that Jimmy can cut out that would leave the rest of the trees looking aesthetically pleasing.

3. Given A = [1, 3, 1, 2] your function should return 0, since the trees are already aesthetically pleasing and no removal is needed.

Assume that:

- N is an integer within the range [4..200];
- each element of array A is an integer within the range [1..1,000].

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

Copyright 2009–2021 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Autocomplete is connected | All changes saved

Test Output

OK

Example test OK

Example test OK

Your code is example test.

@一亩三分地

```
In [67]: 1 def jimmy_tree(A):
2     n = len(A)
3     height_diff = [A[i]-A[i-1] for i in range(1, n)]
4
5     diff_product = [height_diff[i-1]*height_diff[i] for i in range(1, n-1)]
6
7     wrong_idx = [i for i in range(len(diff_product)) if diff_product[i]>=0]
8
9     if len(wrong_idx) == 0:
10         return 0
11     if len(wrong_idx) > 1:
12         return -1
13
14     ans = []
15     for i in range(n):
16         if is_correct_order(A[:i]+A[i+1:]):
17             ans.append(A[:i]+A[i+1:])
18
19     return ans
20
21 def is_correct_order(A):
22     height_diff = [A[i]-A[i-1] for i in range(1, len(A))]
23     diff_product = [height_diff[i-1]*height_diff[i] for i in range(1, len(height_diff))]
24     return max(diff_product) < 0
25
26 print(jimmy_tree([3,4,5,3,7]))
27 print(jimmy_tree([1,2,3,4]))
28 print(jimmy_tree([1,3,1,2]))
29 arr = np.random.randint(1,10, 7).tolist()
30 print(arr, jimmy_tree(arr))
```

```
[[4, 5, 3, 7], [3, 5, 3, 7], [3, 4, 3, 7]]
-1
0
[2, 4, 2, 2, 7, 6, 8] -1
```

**N people**

Task 1

Java 8 English

2 There are N people, numbered from 0 to N-1, playing a game. The K-th person is assigned the letter S[K]. At the beginning the 0th person sends a message, consisting of a single letter S[0], to the A[0]-th person. When the K-th person receives the message, they append their letter S[K] to the message and forward it to A[K]. The game ends when the 0th person receives the message. Find the final message.

You can assume that A contains every integer from 0 to N-1 exactly once.

Write a function:

```
class Solution { public String solution(String S, int[] A); }
```

that given a string S and an array of integers A, both of length N, returns a string denoting the final message received by the 0th person.

Examples:

- Given S = "cdeo" and A = [3, 2, 0, 1], your function should return "code". At the beginning, the 0th person sends a message "c" to the 3rd person. Next, the 3rd person forwards the message "co" to the 1st person. After that the 1st person forwards the message "cod" to the 2nd person. After appending the letter 'e' to it, the 2nd person forward it to the 0th person. The final message, received by 0th person, is "code".
- Given S = "cdeenetpi" and A = [5, 2, 0, 1, 6, 4, 8, 3, 7], your function should return "centipede".
- Given S = "bytdag" and A = [4, 3, 0, 1, 2, 5], your function should return "bat". Notice, that not all letters from S have to be used.

Assume that:

- N is an integer within the range [1..1,000];
- string S consists only of lowercase letters (a-z);
- A contains all integers within range [0..N-1];
- S and A are both of length N.

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

```
In [5]: 1 def n_people(S, A):
2     ans, next_pos = S[0], A[0]
3     while True:
4         ans += S[next_pos]
5         next_pos = A[next_pos]
6         if next_pos == 0:
7             break
8     return ans
9
10 print(n_people('cdeo', [3,2,0,1]))
11 print(n_people('cdeenetpi', [5,2,0,1,6,4,8,3,7]))
12 print(n_people('bytdag', [4,3,0,1,2,5]))
```

code  
centipede  
bat

3. checkers board game 用的recursive

Type Markdown and LaTeX:  $\alpha^2$

Type Markdown and LaTeX:  $\alpha^2$

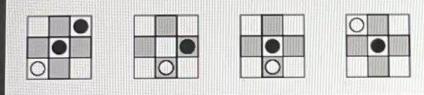
## chess board

wayfair

1 Task 3 Python English Files

2 Jafar can beat one of Aladdin's pawns by moving his pawn two steps in the up-right direction, and then he can make an additional move, beating another of Aladdin's pawns in the up-left direction. After this move there are no more of Aladdin's pawns available to beat, so Jafar's turn ends.

3 In the following situations:



Jafar cannot beat Aladdin's pawn. In the first case, the field two steps in the up-right direction from Jafar's pawn (white) is occupied by one of Aladdin's pawns (black); in the second case, this field is placed outside the board; in the third case, Jafar's pawn cannot move directly upwards; and finally, in the fourth case, Jafar's pawn cannot move in the down-right direction.

What is the maximum number of pawns owned by Aladdin that Jafar can beat in his turn?

Write a function:

```
def solution(B)
```

which, given a square board of  $N \times N$  size describing Aladdin's and Jafar's pawns, returns the maximum number of pawns Jafar can beat in one turn. If none of Aladdin's pawns can be beaten, the function should return 0.

Jafar's pawn is described by the '0' character, Aladdin's pawns by 'X' characters and empty fields by '.' (dots). The board is described from top to bottom and from left to right.

For example, given:

```
B[0] = "...X..."  
B[1] = "......"  
B[2] = "...X..."  
B[3] = "...X..."  
B[4] = "...X.X..."  
B[5] = "...0..."
```

Test Output  
Compilation success  
Example test: [...0...]  
WRONG ANSWER (got 1, expected 2)  
Example test: [...X...]  
WRONG ANSWER (got 1, expected 2)

@一亩三分地

wayfair

1 Task 3 Python English Files

2 which, given a square board of  $N \times N$  size describing Aladdin's and Jafar's pawns, returns the maximum number of pawns Jafar can beat in one turn. If none of Aladdin's pawns can be beaten, the function should return 0.

3 Jafar's pawn is described by the '0' character, Aladdin's pawns by 'X' characters and empty fields by '.' (dots). The board is described from top to bottom and from left to right.

For example, given:

```
B[0] = "...X..."  
B[1] = "......"  
B[2] = "...X..."  
B[3] = "...X..."  
B[4] = "...X.X..."  
B[5] = "...0..."
```

the function should return 2 (Jafar can beat Aladdin's pawn in the up-right direction and then another one in the up-left direction).

Given:

```
B[0] = "X...."  
B[1] = ".X..."  
B[2] = "...0..."  
B[3] = "...X..."  
B[4] = "...."
```

the function should return 0.

Assume that:

- $N$  is an integer within the range  $[1..30]$ ;
- all strings in  $B$  consist only of the following characters: '.', 'X' and/or '0';
- the board is  $N \times N$  square;
- the board contains exactly one pawn owned by Jafar.

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

Copyright 2009–2021 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Test Output  
Compilation success  
Example test: [...0...]  
WRONG ANSWER (got 1, expected 2)  
Example test: [...X...]  
WRONG ANSWER (got 1, expected 2)

@一亩三分地

In [71]:

```
1 def chess(B):
2
3     n = len(B)
4     x_pos_list = []
5     for row in range(n):
6         for col in range(n):
7             if B[row][col] == 'X':
8                 x_pos_list.append((row,col))
9             elif B[row][col] == '0':
10                 o_row,o_col = row,col
11
12     queue = [(o_row,o_col,0)] # (o_row,o_col,count)
13     max_count = 0
14     while queue:
15         o_row,o_col,count = queue.pop()
16         max_count = max(max_count, count)
17         if o_row >= 2 and o_col >= 2 and (o_row-1,o_col-1) in x_pos_list and (o_row-2,o_col-2) in x_pos_list:
18             queue.append((o_row-2,o_col-2,count+1))
19         if o_row >= 2 and o_col+2 < n and (o_row-1,o_col+1) in x_pos_list and (o_row-2,o_col+2) in x_pos_list:
20             queue.append((o_row-2,o_col+2,count+1))
21     return max_count
22
23 print(chess(['..X...', '.....', '....X.', '.X....', '..X.X.', '...0...']))
24 print(chess(['X....', '.X...', '..0..', '...X.', '.....']))
```

2  
0

LC768 <https://leetcode.com/problems/max-chunks-to-make-sorted-ii/>  
[\(https://leetcode.com/problems/max-chunks-to-make-sorted-iii/\)](https://leetcode.com/problems/max-chunks-to-make-sorted-iii/)

1 Task 3

2 We are given an array A consisting of N distinct integers. We would like to sort array A into ascending order using a simple algorithm. First, we divide it into one or more slices (a slice is a contiguous subarray). Then we sort each slice. After that, we join the sorted slices in the same order. Write a function solution that returns the maximum number of slices for which the algorithm will return a correctly sorted array.

3

**Examples:**

1. Given A = [2, 4, 1, 6, 5, 9, 7], the function should return 3. The array can be split into three slices: [2, 4, 1], [6, 5] and [9, 7]. Then, after sorting each slice and joining them together, the whole array will be sorted into ascending order.



2. Given A = [4, 3, 2, 6, 1], the function should return 1. The array cannot be split into smaller slices; it has to be sorted all at once.



3. Given A = [2, 1, 6, 4, 3, 7], the function should return 3.

Autocomplete is connecting...

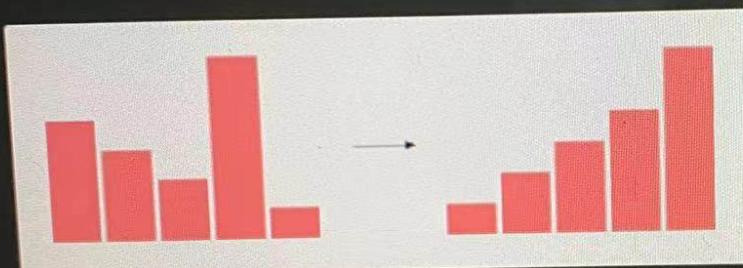


@一亩三分地

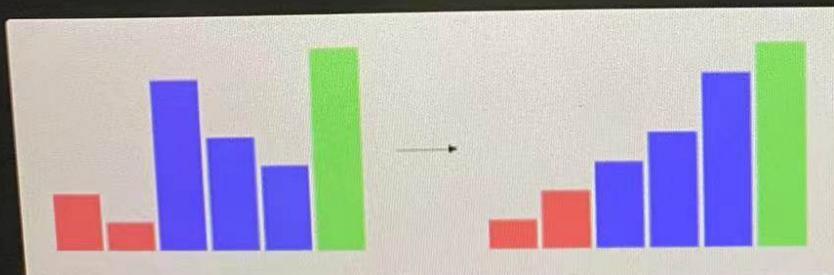
## Task 3



2. Given  $A = [4, 3, 2, 6, 1]$ , the function should return 1. The array cannot be split into smaller slices; it has to be sorted all at once.



3. Given  $A = [2, 1, 6, 4, 3, 7]$ , the function should return 3.



Write an efficient algorithm for the following assumptions:



- $N$  is an integer within the range  $[1..100,000]$ ;
- each element of array  $A$  is an integer within the range  $[1..1,000,000,000]$ ;
- the elements of  $A$  are all distinct.



Copyright 2009–2022 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.



@一亩三分地

Autocomplete is connected | All changes saved

In [ ]:

1

[VO] 整体是一个case study，告诉你wayfair目前遇到的一个问题，和你讨论整个pipeline，从数据的特点开始，

到用什么model，会有什么问题，等等，用的是传统的ml方法，会问比较多细节。面试官人很和善。

[VO] 问的是公司是否应该增加送货上门帮忙组装的服务，中间穿插了简单的计算题。回答说了cost revenue analysis, competitor, 还有结合covid的情况的一些分析。

[VO] 跟以前仓库ood类似，这次是让我设计一个存产品的类，然后添加和查询产品的信息。之后会问如果每个产品属于一个类别，总共有几百万个类别，选择什么数据结构存取最高效。

我其实一开始是从system design的角度讲，这堆东西其实总共也占不了太大地方，给他算了个大概的文件大小，结论是对于这个级别的公司来说最好还是直接全存内存里cache，反正机器他们租得起，不要在意重复。后来他说假设没钱租，我觉得假如一定要省空间的话，就是用tree或者graph和hash相结合来存，这样相同的物品（node）不会重复存太多次，但是维护关系（edge）其实不容易。当然你也可以想想有没有啥更好的方法可以即快又容易的。

### **Amanda buy car**

Amanda wants to buy a car. Her budget is limited, and she has decided that she would prefer to spend it on a car that was produced recently. Her other option is to buy an older, used car, but she would not want to spend much money on such a vehicle. Help her choose all possible options, knowing that she can spend at most 50,000 on a new car produced in 2018 or later; or at most 20,000 on a used car produced in 2010 or later.

You are given a non-empty table, `cars`, whose structure is described by the following query:

```
create table cars (
    id integer not null PRIMARY KEY,
    condition varchar(10) not null,
    year integer not null,
    price integer not null
);
```

Each row of the table `cars` contains information about a car: unique id (`id`), condition (`condition`), year of production (`year`) and price (`price`). Condition contains either one of two possible words: `New` or `Used`.

Write an SQL query that returns a table consisting of one column (`id`), containing the ids of all the cars that fulfil Amanda's criteria. Rows should be ordered by increasing `id`.

**Examples:**

1. Given the following data:

`cars:`

<code>id</code>	<code>condition</code>	<code>year</code>	<code>price</code>
12	New	2018	30000
10	Used	2007	13000
9	Used	2015	10000
1	New	2015	21000
7	Used	2015	23000
5	Used	2012	12000
6	New	2019	150000

your query should return the following table:

```
id
--
5
9
12
```



**senior**

The company you work for has been doing well recently and wants to hire a lot more employees! It is interested in software developers, particularly senior ones. Data about candidates who successfully passed the recruitment process is saved in the table `candidates`.

```
create table candidates (
    id int primary key,
    position varchar not null,
    salary int not null
);
```

Each record in this table contains a unique `id` for each candidate, their `position` (which is either "junior" or "senior") and their negotiated `salary` in dollars per month. Candidates have increasing `ids` and are sorted in ascending order by `salaries`.

Your company's monthly budget for new employees' salaries is 50,000 dollars. It wants to hire as many senior developers as possible, and then use the remaining money to hire as many junior developers as possible.

You were assigned the task of writing a SQL query determining how many employees can be hired for each position. It should return an `employees` table with two columns: `juniors` and `seniors`. The table should contain one row with the number of new employees in each position.

#### Examples:

##### 1. Given table:

id	position	salary
20	junior	10000
30	senior	15000
40	senior	30000

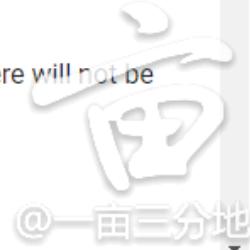
Your query should return:

juniors	seniors
0	2

The company can hire both senior candidates for 45,000 dollars in total, but there will not be enough money left to hire the junior developer candidate.

##### 2. Given table:

id	position	salary
10	senior	40000



## judging system

An automatic judging system checks a solution for the task on multiple test cases. The outcome of a test case is binary: either the solution succeeds or fails. But not all test cases are equally important. Each test case is assigned to a group, and every test case in the group is worth the same number of points.

You have received a raw report with the results of the automatic testing. The report consists of two tables, `test_groups` and `test_cases`, with the following structure:

```
create table test_groups (
    name varchar(40) not null,
    test_value integer not null,
    unique(name)
);

create table test_cases (
    id integer not null,
    group_name varchar(40) not null,
    status varchar(5) not null,
    unique(id)
);
```

Each row of the table `test_groups` contains information @一亩三分地

Each row of the table `test_groups` contains information about a single group of tests: unique name (`name`) and the value of each test in the group (`test_value`). Each row of the table `test_cases` contains information about a single test case: unique id (`id`), the name of the group to which it belongs (`group_name`) and status (`status`). Status contains either one of two possible words: `OK` or `ERROR`.

Write an SQL query that summarizes each group of tests. The table of results should contain four columns: `name` (name of the group), `all_test_cases` (number of tests in the group), `passed_test_cases` (number of test cases with the status `OK`), and `total_value` (total value of passed tests in this group). Rows should be ordered by decreasing `total_value`. In the case of a tie, rows should be sorted lexicographically by `name`.

Examples:

1. Given:

`test_groups`:

name	test value
group A	100
group B	200
group C	300
group D	400
group E	500
group F	600
group G	700
group H	800
group I	900
group J	1000



@一亩三分地

**Examples:**

1. Given:

**test\_groups:**

name	test_value
performance	15
corner cases	10
numerical stability	20
memory usage	10

**test\_cases:**

id	group_name	status
13	memory usage	OK
14	numerical stability	OK
15	memory usage	ERROR
16	numerical stability	OK
17	numerical stability	OK
18	performance	ERROR
19	performance	ERROR
20	memory usage	OK
21	numerical stability	OK

your query should return:

name

| all test cases |

@一亩三分地



your query should return:

	name	all_test_cases	
	passed_test_cases	total_value	
4	numerical stability	4	
2	memory usage	3	
0	corner cases	0	
0	performance	2	

2. Given:

test\_groups:

name	test_value
performance	15
corner cases	10
numerical stability	20
memory usage	10
partial functionality	20
full functionality	40

test\_cases:



1

1

