

This is Google's cache of <https://leetcode.com/discuss/interview-question/801590/>. It is a snapshot of the page as it appeared on Sep 15, 2020 09:33:20 GMT. The [current page](#) could have changed in the meantime. [Learn more](#).

**Full version**   [Text-only version](#)   [View source](#)

Tip: To quickly find your search term on this page, press **Ctrl+F** or **⌘-F** (Mac) and use the find bar.



Explore <sup>Day 20</sup>

Problems



August LeetCode Coding Challenge

Mock

Contest

Discuss



Store

Premium



Sign  
up

or

Sign  
in

< Back

Amazon | OA2 | Pagination



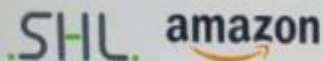
Anonymous User

Last Edit: 14 hours ago   2.9K VIEWS

13

Position: New Grad 2021



**Problem** | Test Cases | Output

Within the Amazon website, we have many products to choose from. Therefore, in order to help our customers quickly find what they are looking for, we provide the ability to search, sort, and filter the results. Because we have so many products, however, they don't all fit on one page. To address this we need a pagination solution! The search results from our search engine come back as an unordered list of 0 to many items. Each item has its name, relevance and price. Before sending this back to the client we need to sort it based on one of any of the defined attributes and then, to prevent sending too much data to the client, we need to break the result up into pages and return only the requested page.

Given a list of items, the sort column, the sort order (0: ascending, 1: descending), the number of items to be displayed in each page, and a page number, write an algorithm to determine the list of item names in the specified page while respecting the item's order (Page number starts at 0).

**Input**

The input to the function/method consists of six arguments:

*numOfItems*, an integer representing the number of items;

*items*, a map with item names as keys with a list of pair of integers - relevance and price of the item;

*sortParameter*, an integer representing the value used for sorting (0 for name, 1 for relevance, 2 for price);

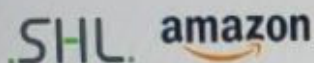
*sortOrder*, an integer representing the order of sorting (0 for ascending order and 1 descending order);

*itemsPerPage*, an integer representing the number of items to be displayed per page;

*pageNumber*, an integer representing the page number to display item names.

**Output**

Return a list of strings representing the list of item names on the requested page in the order they are displayed.

**Note****Problem** | Test Cases | Output**Note**

*itemsPerPage* is always greater than 0 and is always less than the minimum of *numOfItems* and 20.

**Constraints**

$1 \leq \text{numOfItems} < 10^5$

$0 \leq \text{relevance, price} < 10^8$

$0 \leq \text{pageNumber} < 10$

**Example**

Input:

```
numOfItems = 3  
items =  
[["item1", 10, 15],  
["item2", 3, 4],  
["item3", 17, 8]]  
sortParameter = 1  
sortOrder = 0  
itemsPerPage = 2  
pageNumber = 1
```

Output:

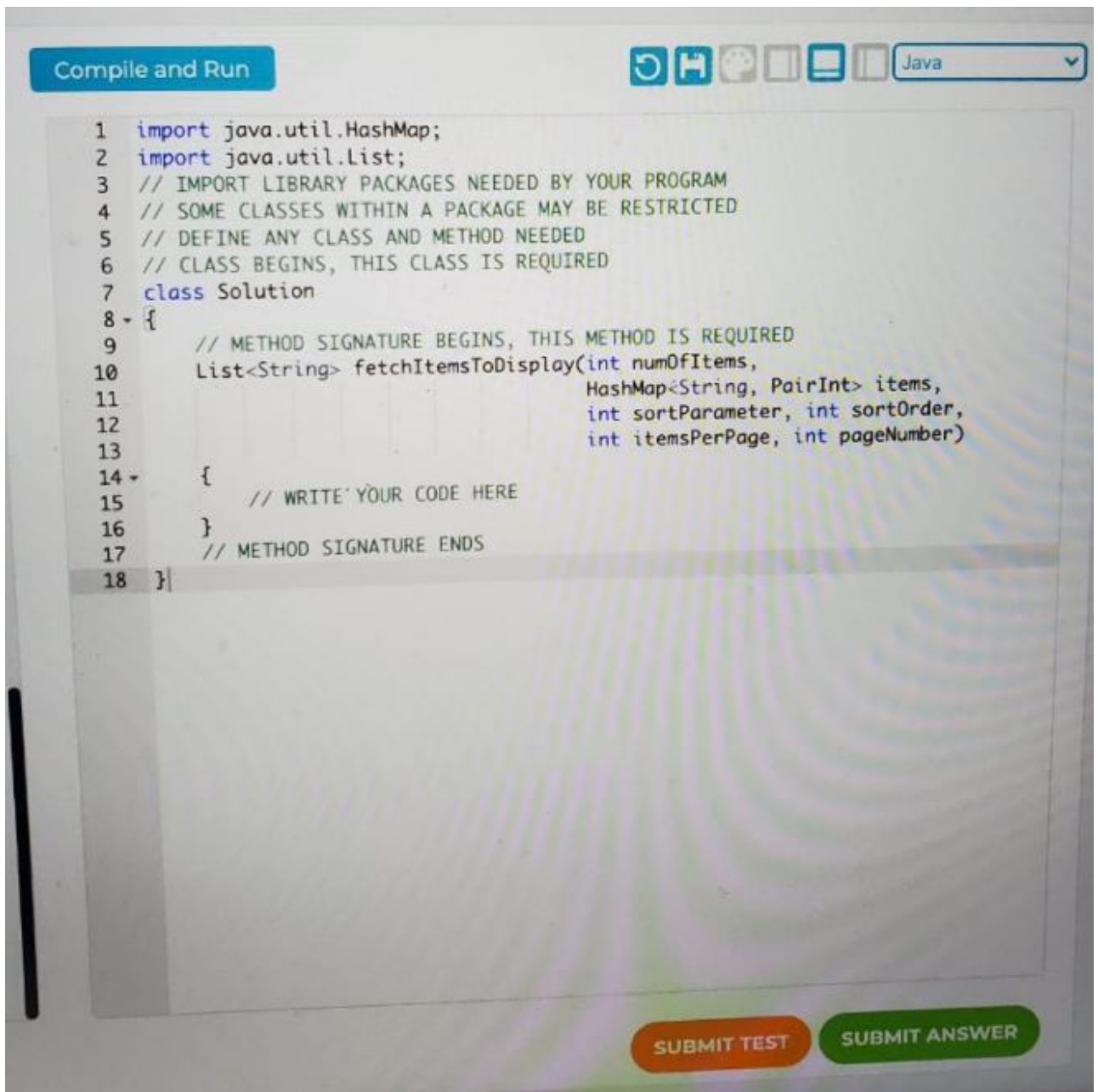
```
["item3"]
```

Explanation:

There are 3 items. Sort them by (relevance : 1) in ascending order ( *items* = [ ["item2", 3, 4], ["item1", 10, 15], ["item3", 17, 8]] ). Display up to 2 items on each page. The page 0 contains 2 item names ["item2", "item1"] and page 1 contains only 1 item name, result = "item3".

**Helper Description**

The following class is used to represent a Pair of integers which is already implemented in



```
1 import java.util.HashMap;
2 import java.util.List;
3 // IMPORT LIBRARY PACKAGES NEEDED BY YOUR PROGRAM
4 // SOME CLASSES WITHIN A PACKAGE MAY BE RESTRICTED
5 // DEFINE ANY CLASS AND METHOD NEEDED
6 // CLASS BEGINS, THIS CLASS IS REQUIRED
7 class Solution
8 {
9     // METHOD SIGNATURE BEGINS, THIS METHOD IS REQUIRED
10    List<String> fetchItemsToDisplay(int numItems,
11                                    HashMap<String, PairInt> items,
12                                    int sortParameter, int sortOrder,
13                                    int itemsPerPage, int pageNumber)
14    {
15        // WRITE YOUR CODE HERE
16    }
17    // METHOD SIGNATURE ENDS
18 }
```

Can somebody solve this ?

Comments: 14

☒ Best ☐ Most Votes ☐ Newest to Oldest ☐ Oldest to Newest

Login to Comment



Ohileshwar ★ 8 a day ago

PriorityQueue for the win

▲ 5 ▼ ↩ Reply



Bhavik128 ★ 10 a day ago

[Read More](#)

▲ 8 ▼  Show 4 replies  Reply



kamui\_amaterasu33  
[kamui\\_amaterasu33's](#)

★ 68

Last Edit: 10 hours ago

[avatar](#)[Read More](#)

▲ 2 ▼  Show 4 replies  Reply



gerrob

★ 428

Last Edit: 21 hours ago

So far nobody came up with the optimal solution. That is using quickselect to find the first and the last item to display in only  $O(n)$  time and in another  $O(n)$  time select the displayed items, then sort them with any fast sorting algorithm. The time will be  $O(k * \log(k) + n)$  and this running time is optimal, where  $k$  is the number of items needed to display on the requested page.

▲ 2 ▼  Show 1 reply  Reply



celmar\_stone

★ 6

a day ago

What about this python solution:

```
class Solution:
    def get_items(self, n, items, sortParameter, sortOrder, itemsPerPage, pageNumber):
        items.sort(key=lambda x: x[sortParameter], reverse=sortOrder)
        return items[pageNumber*itemsPerPage: (pageNumber+1)*itemsPerPage]
```

▲ 3 ▼  Show 2 replies  Reply



kamui\_amaterasu33

★ 68

Last Edit: an hour ago

[kamui\\_amaterasu33's](#)[avatar](#)[Read More](#)

▲ 1 ▼

Reply



innerpieces

★ 4

Last Edit: a day ago

[Read More](#)

▲ 1 ▼

Show 1 reply Reply



viditkulshreshtha26

★ 3

3 minutes ago

If you don't mind, can you please share the other question too. It would be of great help

▲ 0 ▼

Reply



Juandollaa

★ -33

Last Edit: 3 hours ago

[Read More](#)

▲ 0 ▼  Reply



buttcheeks ★ 126 13 hours ago

similar problem on leetcode ?

▲ 0 ▼  Reply

□ 1 2 □

