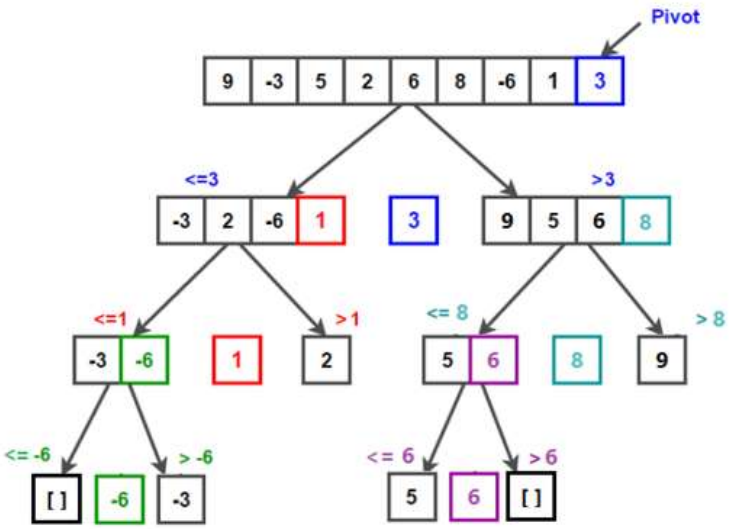
Ayo Quickselect

Author: liemroym

Mencari nilai minimum ke X merupakan kasus yang mudah apabila data yang dimiliki sedikit atau sudah tersortir. Namun, apabila data cukup banyak dan tidak tersortir, maka pencarian nilai minimum ke X tentunya menjadi sulit untuk dilakukan. Kompleksitas algoritma sorting tercepat saat ini; quicksort adalah O(n log n) yang sudah tergolong cepat bila dibandingkan algoritma lainnya. Namun untuk kasus pencarian nilai minimum ke X, penggunaan algoritma quicksort cenderung redundan karena kedua interval hasil pemisahan pivot ditelusuri dan disatukan kembali (divide and conquer), sedangkan nilai minimum ke X dapat ditemukan hanya dekan menelusuri salah satu intervalnya saja:



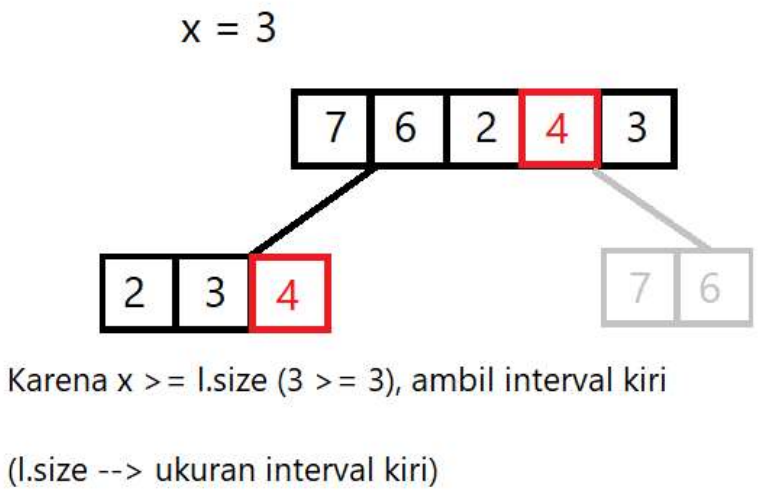
Maka dari itu, terdapat satu algoritma pencarian nilai minimum ke X yang menggabungkan konsep quicksort dan decrease and conquer, yaitu algoritma Quickselect. Dalam algoritma ini, setelah array dipisah berdasarkan pivot, akan dilakukan pemilihan interval untuk ditelusuri berdasarkan nilai X dan interval lainnya tidak digunakan. Contohnya:

- 7 6 2 4 3, x = 3
- Pivot random: 4
- Left array (<= pivot): 2 3 4 **(pivot dimasukkan lagi ke interval)**

- Right array (> pivot): 7 6
- Elemen dengan nilai kecil pasti berada di left array
- Karena x = 3 dan ukuran left array = 3, elemen minimum ke x pasti terletak di left array, sehingga right array tidak perlu ditelusuri (decrease and conquer)

Algoritma Quickselect memiliki kompleksitas waktu rata-rata O(n), yang lebih cepat dibandingkan melakukan sorting terlebih dahulu (O(n log n)). Namun, algoritma ini tidak konsisten, karena bergantung pada pivot yang dipilih. Pada worst case nya, algoritma ini memiliki kompleksitas O(n^2), yang lebih lambat dibandingkan melakukan sorting terlebih dahulu

Contoh lengkap dapat dilihat sebagai berikut (pivot random dan diletakan di left array):



Implementasikan algoritma Quickselect untuk mencari nilai minimum ke X! Terdapat beberapa test case yang tidak dapat diselesaikan hanya dengan sorting (karena time complexity), sehingga pastikan untuk menggunakan algoritma Quickselect!

Note: gunakan pivot random. Kompleksitas waktu algoritma Quickselect bergantung pada pivot yang dipilih, sehingga apabila pada percobaan pertama terkena error time limit, **ulangi submit hingga tidak terjadi error (pivot berubah-ubah pada setiap submit)**. Ingat, nilai yang diambil dalam setiap submit di Hackerrank adalah **nilai tertinggi**

Input Format

Baris pertama berisikan n dan x

Baris kedua berisikan n buah integer dipisahkan dengan spasi yang merupakan elemen-elemen array (arr[i]). Array tidak terurut dan dipastikan bahwa tidak ada elemen yang sama dalam satu array

Constraints

1 <= n <= 10^7

1 <= x <= n

-10^7 <= arr[i] <= 10^7

Output Format

Outputkan satu buah integer yang merupakan nilai minimum terkecil ke-x dari array arr

Sample Input 0

```
5 3
7 6 2 4 3
```

Sample Output 0

```
4
```

Contest ends in 2 days

Submissions: 66
Max Score: 100
Difficulty: Medium

Rate This Challenge:

More

Python 3

```python
import random

def quickselect(arr, x):
    # Jika array hanya memiliki satu elemen, return elemen
    if len(arr) == 1:
        return arr[0]

    # pivot random
    pivot = random.choice(arr)

    # Pisahkan array menjadi left dan right array berdasarkan pivot
    left = [elem for elem in arr if elem <= pivot]
    right = [elem for elem in arr if elem > pivot]

    # Jika jumlah elemen di left array sama dengan x, maka elemen terakhir dari left array adalah elemen minimum ke-x
    if len(left) == x:
        return max(left)
    # Jika jumlah elemen di left array lebih kecil dari x, maka cari elemen minimum ke-x pada right array
    elif len(left) < x:
        return quickselect(right, x - len(left))
    # Jika jumlah elemen di left array lebih besar dari x, maka cari elemen minimum ke-x pada left array
    else:
        return quickselect(left, x)


# Main program
n, x = map(int, input().split())
arr = list(map(int, input().split()))

print(quickselect(arr, x))
```

31

Line: 1 Col: 1

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