
Los K elementos más frecuentes

Z12480_es

Dada una lista de números enteros **nums** y un número entero **k** (que debe ser menor o igual al número de elementos diferentes que hay en **nums**) escribir una función:

k_mas_frecuentes(nums, k)

que devuelva una lista con los **k** elementos que aparecen con más frecuencia dentro de la lista **nums**. Podéis devolver la respuesta en cualquier orden.

No se permite ordenar *nada* (ni la lista **nums**, ni ninguna otra estructura de datos auxiliar que se utilice) *dentro* de la función requerida. Vuestra propuesta de solución de la función **k_mas_frecuentes(nums, k)** no puede usar `sort`, `sorted` ni variantes.

Parámetros y retorno de la función solicitada

Los parámetros de la función **k_mas_frecuentes(nums, k)** son una lista de enteros **nums** y un número **k** que podemos suponer que es menor o igual que el número de elementos diferentes que hay en **nums**.

La función **k_mas_frecuentes(nums, k)** debe devolver una lista con los **k** números más frecuentes en la lista **nums**. No importa el orden en que aparezcan, que el programa principal en **code.py** ya se ocupa de ordenar antes de escribir.

Entrada

La entrada al programa será, en este orden, el número **k**, un número **n** y después una colección de **n** números enteros.

Véanse los ejemplos del juego de pruebas público.

Salida

El programa debe escribir una lista con los **k** elementos más frecuentes de la lista de **n** números proporcionados.

Véanse los ejemplos del juego de pruebas público.

Observaciones

Debéis descargaros el archivo **code.py** (icono de la serpiente). Este archivo es un programa con **todo** lo necesario para ejecutar los juegos de prueba públicos. Solo falta, claro, la función que os pide el enunciado. Este archivo debéis completarlo con el código que falta, y eso, **todo**, es lo que debéis enviar al Jutge como solución.

Dentro del archivo **code.py** tenéis la clase **Heap** que hemos trabajado en las clases de laboratorio. No será necesario que vuestra solución haga ningún *import* ni nada. Todo el código que necesitáis lo tenéis dentro de **code.py**.

En los juegos de prueba de este problema, tanto públicos como privados, os garantizamos que la lista de los **k** elementos más frecuentes solicitada es **única** (excepto permutaciones).

La eficiencia y la calidad de la solución se tendrán en cuenta en la corrección manual. No se puede utilizar la función `sorted`, ni el método `sort` ni variantes. Si se ordena *cualquier cosa* dentro de la función requerida el estudiante será penalizado y la solución se considerará incorrecta.

Ejemplo de entrada 1

5
17
4 1 3 2 8 7 7 10 5 3 2 7 4 5 2 5 5

Ejemplo de salida 1

[2, 3, 4, 5, 7]

Ejemplo de entrada 2

3
18
5 7 5 1 3 3 3 10 9 7 3 3 2 4 8 6 5 5

Ejemplo de salida 2

[3, 5, 7]

Ejemplo de entrada 3

6
19
6 2 10 3 6 6 2 3 9 3 3 8 10 5 9 4 10 3 5

Ejemplo de salida 3

[2, 3, 5, 6, 9, 10]

Ejemplo de entrada 4

4
16
7 1 6 2 6 3 8 4 1 8 5 4 9 6 4 8

Ejemplo de salida 4

[1, 4, 6, 8]

Ejemplo de entrada 5

3
20
6 7 10 4 5 4 3 10 5 8 7 9 1 3 3 3 6 3 4 6

Ejemplo de salida 5

[3, 4, 6]

Ejemplo de entrada 6

1
8
6 7 2 9 7 4 2 2

Ejemplo de salida 6

[2]

Información del problema

Autoría: Jordi Delgado

Generación: 2026-04-01T05:21:46.087Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>