471. Top K Frequent Words

* [Description](http://www.lintcode.com/en/problem/top-k-frequent-words/" \l "description)
* [Notes](http://www.lintcode.com/en/problem/top-k-frequent-words/#note)
* [Testcase](http://www.lintcode.com/en/problem/top-k-frequent-words/#testcase)
* [Judge](http://www.lintcode.com/en/problem/top-k-frequent-words/#judge)

Given a list of words and an integer k, return the top k frequent words in the list.

 Notice

You should order the words by the frequency of them in the return list, the most frequent one comes first. If two words has the same frequency, the one with lower alphabetical order come first.

Have you met this question in a real interview?

Yes

**Example**

Given

[

"yes", "lint", "code",

"yes", "code", "baby",

"you", "baby", "chrome",

"safari", "lint", "code",

"body", "lint", "code"

]

for k = 3, return ["code", "lint", "baby"].

for k = 4, return ["code", "lint", "baby", "yes"],

[**Challenge**](http://www.lintcode.com/en/problem/top-k-frequent-words/#challenge)

<http://www.lintcode.com/en/problem/top-k-frequent-words/#>

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication18;

import java.util.\*;

import java.util.Map.Entry;

/\*\*

\*

\* @author Usuario

\*/

public class JavaApplication18 {

public static String[] topKFrequentWords(String[] words, int k) {

// write your code here

if(k == 0) return new String[0];

HashMap<String, Integer> hm = new HashMap<>();

//HashMap<String, Integer> hm =

// new HashMap();

for(int i =0; i<words.length; i++) {

if(hm.containsKey(words[i])){

hm.put(words[i], hm.get(words[i])+1);

} else {

hm.put(words[i], 1);

}

}

TreeMap<Integer, ArrayList< String>> ord =

new TreeMap();

for(String key : hm.keySet()) {

if(ord.containsKey(hm.get(key))) {

ArrayList<String> aux= ord.get(hm.get(key));

aux.add(key);

//Collections.sort(aux);

ord.put(hm.get(key),aux );

} else {

ArrayList<String> aux = new ArrayList();

aux.add(key);

//Collections.sort(aux);

ord.put(hm.get(key), aux);

}

}

TreeMap<Integer, ArrayList< String>> sorted =

new TreeMap(Collections.reverseOrder());

for(int key : ord.keySet()) {

ArrayList<String> aux = ord.get(key);

Collections.sort(aux);

sorted.put(key, aux);

}

int cont =0;

String[] topk = new String[k];

int indice =0;

for(int key : sorted.keySet()) {

ArrayList<String> aux = sorted.get(key);

//System.out.print(key + " ");

for(int i =0; i<aux.size(); i++) {

// System.out.print(aux.get(i) + " ");

topk[indice++] =aux.get(i);

if(indice >= k) {

return topk;

}

}

//System.out.println();

}

return topk;

}

public static void main(String[] args) {

// TODO code application logic here

String[] words = {

"yes", "lint", "code",

"yes", "code", "baby",

"you", "baby", "chrome",

"safari", "lint", "code",

"body", "lint", "code"

}; //["code", "lint", "baby"]

int k=3;

// String[] words = {"a","b","c","a","c"};

// int k =2;

for(String elem: topKFrequentWords(words, k)) {

System.out.print(elem + " ");

}

/\*

for(String elem : res) {

System.out.print(elem + " ");

}\*/

}

}