
Suma dels k més petits

P86160_ca

Els Jordis (Cortadella i Petit, no els de debò) tenen la conversa següent:

- Necessitem una funció `int sum_smallest (vector<int> v, int k)` que, donat un vector v amb n enters i un valor $k \in [0..n]$, retorni la suma dels k valors més petits de v .
- Vols dir, per exemple, que per $v = \{4, 91929, 2, 4, 1\}$ i $k = 3$ retorni 7?
- Exacte!
- Val. Umhhh... Crec que ja ho tinc: Ordenarem el vector amb un quick sort i després calcularem la suma de les seves k primeres posicions.
- És clar! A les transpes d'AP2 i a lliçons.jutge.org ja hi tenim una implementació del quick sort amb la cèlebre partició de Hoare. Amb un *cut and paste* ho tindrem solucionat en un momentet.

[10 minuts més tard, el codi complet està llest; vegeu-lo al darrere.]

- Mira, ja ho tinc fet!
- Fantàstic, però quin és el cost d'aquest algorisme?
- Doncs... la fase d'ordenació amb quick sort és $O(n \log n)$ en mitjana i la fase de suma és $O(n)$ perquè $k \leq n$. Per tant, el cost total és $O(n \log n)$ en mitjana.
- Correcte! Però... cal ordenar tot el vector? No es podria fer més eficient?

Modifiqueu el programa dels Jordis per tal que la funció `sum_smallest()` tingui cost $O(n)$ en mitjana.

Observació

Només cal enviar el procediment demanat; el programa principal serà ignorat.

Als jocs de prova del Jutge, els vectors v seràn molt grans i els seus valors seràn generats aleatòriament.

```

#include <cassert>
#include <iostream>
#include <vector>
using namespace std;

// Hoare's partition.
// Partitions  $v[l..r]$  into  $v[l..q, q+1..r]$  where  $q$  is the returned value
// so that all elements in  $v[l..q]$  are smaller or equal than all elements in  $v[q+1..r]$ .
int partition ( vector<int>& v, int l, int r )
{
    const int x = v[l];
    int i = l - 1;
    int j = r + 1;
    for (;;) {
        while (x < v[--j])
            ;
        while (v[++i] < x)
            ;
        if (i ≥ j)
            return j;
        swap(v[i], v[j]);
    }
}

// Sorts  $v[l..r]$  using quick sort.
void quicksort ( vector<int>& v, int l, int r )
{
    if (l < r) {
        const int q = partition (v, l, r);
        quicksort (v, l, q);
        quicksort (v, q + 1, r);
    }
}

// Returns the sum of the  $k$  smallest elements of  $v$ .
int sum_smallest ( vector<int> v, int k )
{
    int n = v.size ();
    assert (k ≥ 0 and k ≤ n);
    quicksort (v, 0, n - 1);
    int s = 0;
    for (int i = 0; i < k; ++i)
        s += v[i];
    return s;
}

// Just a test
int main()
{
    cout << sum_smallest({ 4, 91929, 2, 4, 1 }, 3) << endl;
}

```

Informació del problema

Autor : Jordi Petit

Generació : 2025-05-13 12:02:37

© *Jutge.org*, 2006–2025.

<https://jutge.org>