```c
/*
 * This code is from "Algorithms in C, Third Edition,"
 * by Robert Sedgewick, Addison Wesley Longman, 1998.
 * (Adapted by Y. Kohayakawa)
 */

#include "Item.h"
#define M 10

int partition(Item a[], int l, int r)
  { int i = l-1, j = r; Item v = a[r];
    for (;;)
      {
        while (less(a[++i], v)) ;
        while (less(v, a[--j])) ;  // no median-of-three, temos uma sentinela
                                   // portanto podemos omitir o
                                   // if (j==l) break;
                                   // Isto faz diferenca no desempenho!
        if (i >= j) break;
        exch(a[i], a[j]);
      }
    exch(a[i], a[r]);
    return i;
  }

void quicksort(Item a[], int l, int r)
  { int i;
    if (r-l <= M) return;
    exch(a[(l+r)/2], a[r-1]);  // necessidade desta troca 'e sutil
                               // Veja exercicio 7.33 de Sedgewick
    compexch(a[l], a[r-1]);
      compexch(a[l], a[r]);
        compexch(a[r-1], a[r]);
    i = partition(a, l+1, r-1);
    quicksort(a, l, i-1);
    quicksort(a, i+1, r);
  }

void insertion(Item a[], int l, int r)
  { int i;
    for (i = l+1; i <= r; i++) compexch(a[l], a[i]);
    for (i = l+2; i <= r; i++)
      { int j = i; Item v = a[i];
        while (less(v, a[j-1]))
          { a[j] = a[j-1]; j--; }
        a[j] = v;
      }
  }

void sort(Item a[], int l, int r)
  {
    quicksort(a, l, r);
    insertion(a, l, r);
  }
```