Examen de Prácticas AA Junio. 2010-2011

Alumno:
Grupo de Teoría:
Grupo de Prácticas:

Compila	Ejecuta	Gráfica	Cuestiones	Memoria	Total

Cuestión a desarrollar sobre la Práctica 2

Objetivo

Introsort (o *introspective sort*) es un algoritmo de ordenación que empieza con *quicksort* y cambia a *mergesort* cuando la profundidad de la recursión excede un nivel calculado como el logaritmo del número de elementos a ordenar.

Pasos a seguir

1. Añadid las siguientes declaraciones a ordena.h:

   ```
   int introsort1(int* tabla, int ip, int iu);
   int introsort(int* tabla, int ip, int iu, pfunc_pivote pivote, int prof);
   ```

   La función introsort tiene un parametro prof que se va decrementando hasta alcanzar 0, en ese momento, cambia de quicksort a mergesort.

2. Implementad en ordena.c las nuevas funciones. La implementación de introsort1 será:

   ```
   int introsort1(int* tabla, int ip, int iu)
   {
     return introsort(tabla, ip, iu, aleat_num, floor(log(iu – ip + 1)/log(2)));
   }
   ```

3. Cread el programa examen.c (similar a P1_5_??.c) que genere los tiempos de ordenación para 1000 permutaciones de tamaños entre 1000 y 5000 con incremento 100.

4. Con el experimento anterior, generad una gráfica y compararla con la de vuestro *quicksort* original. Responder a las siguientes cuestiones:

   a) ¿Son significativamente distintos los tiempos y el número de OBs?
   b) ¿Por qué?
   c) ¿Por qué se elige $\log_2(n)$ como límite de profundidad?

Material a entregar

1. El código de las funciones implementadas y el programa de prueba .

2. Los archivos de texto con el resultado de ejecutar los experimentos, y las respuestas al apartado 4.

3. Un .tar.gz con el grupo y el nombre del alumno que se mandará vía web dentro del grupo correspondiente. (Ej: EXAMEN_grupo_nombre.tar.gz)

NOTA: Si desde la entrega de la práctica o durante el examen se ha realizado alguna mejora en el código indicadlo sobre la hoja de examen.