ANÁLISIS Y DISEÑO DE ALGORITMOS

Divide y Vencerás

Práctica 5 de laboratorio

Entrega: Hasta el domingo 11 de marzo, 23:55h. A través de Moodle

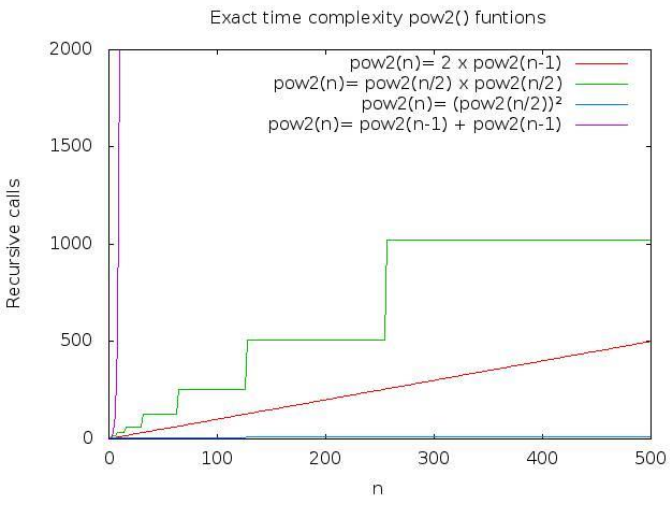
Las siguientes recurrencias matemáticas obtienen el valor de la potencia enésima de 2 (2^n donde $n \in \mathbb{N}$). Se diferencian entre ellas en la división del problema en subproblemas y en la posterior forma de combinar las soluciones parciales.

pow2_1(n)	pow2_2(n)
$\text{pow2\_1(n)} = \begin{cases} 1 & \text{si } n = 0 \\ 2 \times \text{pow2\_1(n} - 1) & \text{si } n > 0 \end{cases}$	$\text{pow2\_2(n)} = \begin{cases} 1 & \text{si } n = 0 \\ \text{pow2\_2(n/2)} \times \text{pow2\_2(n/2)} & \text{si } n \text{ es par} \\ 2 \times \text{pow2\_2(n/2)} \times \text{pow2\_2(n/2)} & \text{si } n \text{ es impar} \end{cases}$
pow2_3(n)	**pow2_4(n)**
$\text{pow2\_3(n)} = \begin{cases} 1 & \text{si } n = 0 \\ r \times r & \text{donde } r = \text{pow2\_3(n/2); si } n \text{ es par} \\ 2 \times r \times r & \text{donde } r = \text{pow2\_3(n/2); si } n \text{ es impar} \end{cases}$	$\text{pow2\_4(n)} = \begin{cases} 1 & \text{si } n = 0 \\ \text{pow2\_4(n} - 1) + \text{pow2\_4(n} - 1) & \text{si } n > 0 \end{cases}$

El objetivo de esta práctica es comprobar empíricamente cómo influye la división en subproblemas en la eficiencia del algoritmo resultante. Para ello se pide:

1. Implementar las funciones incorporándolas a un único código fuente y escribir una función `main()` para obtener una tabla similar a la que se muestra. Contiene el número de llamadas recursivas que realiza cada una de las funciones para distintos valores de n. Por lo tanto, cada función debe calcular también el número de llamadas recursivas que realiza, para ello puedes utilizar variables globales. En cuanto a los valores de n, se debe tomar los números pares desde 0 hasta 500, salvo en el caso de `pot2_4` que por su elevado coste computacional sólo se debe considerar hasta el valor $n = 20$.

n	pow2_1	pow2_2	pow2_3	pow2_4
0	1	1	1	1
2	3	7	3	7
4	5	15	4	31
6	7	15	4	127
8	9	31	5	511
10	11	31	5	2047
12	13	31	5	8191
14	15	31	5	32767
16	17	63	6	131071
18	19	63	6	524287
20	21	63	6	2097151
22	23	63	6	
24	25	63	6	
26	27	63	6	
28	29	63	6	
30	31	63	6	
32	33	127	7	
34	35	127	7	
36	37	127	7	
38	39	127	7	
40	41	127	7	
...	...	...	...	



2. Mediante *Gnuplot* obtén la gráfica asociada a dicha tabla (similar a la de la figura mostrada).

3. Analiza las gráficas obtenidas y saca conclusiones sobre cómo influye en la complejidad temporal la forma de dividir el problema en subproblemas.

Nombres de archivos

Los nombres de los archivos a realizar (no todos se entregan) serán los siguientes:

- `pow2.cc`. Código fuente para obtener la tabla mencionada.
- `pow2.gnuplot`. Fichero de órdenes de *Gnuplot* para obtener la gráfica.
- `pow2.recursiveCalls`. Tabla similar a la mostrada. **Este archivo no debe entregarse**, debe obtenerse a través de la orden `make`.
- `pow2.png`. Gráfica similar a la mostrada. **Este archivo no debe entregarse**, debe obtenerse a través de la orden `make`.
- `makefile`. Archivo similar a los entregados en prácticas anteriores. Debe contener los siguientes objetivos (preferiblemente por este orden):
  1. `pow2`, para crear el ejecutable.
  2. `pow2.recursiveCalls`, para obtener la tabla que contiene el número de llamadas recursivas que cada función ha realizado para cada valor de n analizado.
  3. `pow2.png`, para obtener, a partir del fichero de órdenes de Gnuplot y de la anterior tabla, la gráfica asociada a los resultados obtenidos.
  4. `all`, para llamar a los tres objetivos anteriores.

Normas para la entrega.

ATENCIÓN: Estas normas son de obligado cumplimiento para que esta práctica sea evaluada.

1. Se debe entregar únicamente los ficheros `pow2.cc`, `pow2.gnuplot` y `makefile`. Sigue escrupulosamente los nombres de ficheros, objetivos, etc. que aquí se citan. No hay que entregar nada más.

2. Es imprescindible que no presente errores ni de compilación ni de interpretación (según corresponda), en los ordenadores del laboratorio asignado.[1] Se tratará de evitar también cualquier tipo de *warning*.

3. Todos los ficheros que se entregan deben contener el nombre del autor y su DNI (o NIE) en su primera línea (entre comentarios apropiados según el tipo de archivo).

4. Se comprimirán en un archivo *.tar.gz* cuyo nombre será el DNI del alumno, compuesto de 8 dígitos y una letra (o NIE, compuesto de una letra seguida de 7 dígitos y otra letra). Por ejemplo: 12345678A.tar.gz o X1234567A.tar.gz. **Solo se admite este formato de compresión y solo es válido esta forma de nombrar el archivo.**

5. En el archivo comprimido **no debe existir subcarpetas**, es decir, al extraer sus archivos estos deben quedar guardados en la misma carpeta donde está el archivo que los contiene.

6. La práctica hay que subirla a *Moodle* respetando las fechas expuestas en el encabezado de este enunciado.

[1] Si trabajas con tu propio ordenador o con otro sistema operativo asegúrate de que este requisito se cumple.