Algoritmos y Estructuras de Datos II, Segundo del Grado de Ingeniería Informática, Test de Análisis de Algoritmos, marzo 2014

Para cada problema habrá que justificar razonadamente la respuesta que se dé. No son respuestas "razonadas" algunas como: "porque hay otro problema igual en los apuntes", "porque es evidente", "porque lo he comprobado con varios valores"...

1) (3.5 puntos) Dado el código:

```
for i=1 to n
    if(par(i))
        for j = i + 1 to n
            if(a[i, j] > a[j, i])
                temp = a[i, j]
                a[i, j] = a[j, i]
                a[j, i] = temp
            endif
        endfor
    else
        for j = 1 to i − 1
            if(a[i, j] > a[j, i])
                temp = a[i, j]
                a[i, j] = a[j, i]
                a[j, i] = temp
            endif
        endfor
    endif
endfor
```

donde a:array[1..n,1..n], estudiar los órdenes O, Omega, exacto y o-pequeño de su tiempo de ejecución en el caso más favorable, más desfavorable y promedio.

2) (2 puntos) Resolver las ecuaciones de recurrencia:

a) $t(n) = 2t(n-1) + 4t(n-2) + 2^n$. b) $t(n) = 2t(n-1) + 4t(n-2) + 2n$. c) $t(n) = 2t(n/2) + 4t(n/4) + 2$. d) $t(n) = 2t(n/2) + 4t(n/4) + 2n$

e) $t(n) = 3t(n/2) + 1 + 2^n$

3) (1.5 puntos) Ordenar los siguientes órdenes de ejecución: $O(\log 3^n)$, $O(\log(3^n + n))$, $O(n + \log 3^n)$, $O\left(\log (3^n)^2\right)$, $O\left(\log 3^{(n^2)}\right)$.

4) (3 puntos) Dada la función:

```
int funcion(int a[],int pos):
    if(pos ≤ 20)
        cont = 0
        for i = 1 to pos
            if(a[i] < a[0])
                cont + +
            endif
        endfor
    else if(a[pos] < a[0])
        return funcion(a,pos/2)+1
    else
        return funcion(a,pos/4)
    endif
```

donde a:array[0..n], y suponiendo que la primera llamada se hace con $pos = n$, estudiar los órdenes O, Omega, exacto y o-pequeño de su tiempo promedio de ejecución. Analizar los mismos órdenes para la ocupación promedio de memoria suponiendo que en cada llamada a la función se pasa el array por referencia y suponiendo que se hace una copia del array.