Examen julio 2011

1.- Tipo test.

1:c) 2.-b) 3.-a) 4:d) 5.-c) 6.-b) 7.-a) 8.-b) 9.-b) 10.-a)

2.- La complejidad espacial estima la cantidad de memoria ocupada por un algoritmo mientras que la temporal estima el tiempo de ejecución, normalmente en el peor de los casos.

3.-

- Memoria contigua

-Ventajas: ocupa menos espacio/memoria porque el tamaño es conocido previamente, es más sencillo acceder a los elementos del array.

- Desventajas: se tiene que saber el tamaño previamente y hay limitaciones o se dificulta con las funciones de modificación de la lista (inserción, eliminación).

Las más complicadas son la inserción de nuevos elementos o la eliminación de otros porque también se modifica el tamaño de la lista y hay que mover los elementos para completar los huecos vacíos, etc.

Las más sencillas son acceder a un elemento e insertar uno nuevo al final.

4.- → No tenemos el código porque no se dio.

5.- f_1 → 3 bucles { 0 a 10.000, 0 a num, 0 a num } → $10000n^2$ → $O(n^2)$ { $\Omega(N^2)$ porque los bucles se van a hacer siempre

f_2 → 2 bucles → $O(n^2)$

Programa 4

- Si N%55 == 0 → f_1 → $O(N^2)$
- Si no, → f_2 → $O(N^2)$

Mejor caso → $\Omega(N^2)$
Peor caso → $O(N^2)$

Programa B

f_1 → alg recursivo con 2 llamadas → $O(2^n)$

Peor Caso → $O(2^n)$
Mejor Caso → $O(1)$ → porque si n=1 o n=2, termina directamente.

Programa C

f_1 → 2 bucles (anidados) → n x n → $O(n^2)$ / $\Omega(n^2)$

f_2 → 1 bucle → n → $O(n)$ / $\Omega(n)$

Main →
- Mejor caso → k <= 1000 → solo llama a f_2 → $\Omega(n)$
- Peor caso → k > 1000 → llama a f_2 y f_1 → $O(n+n^2)$ → $O(n^2)$

6.- Producto y potencia:

Sintaxis:

producto (Natural, Natural) → Natural
potencia (Natural, natural) → Natural

Semántica: ∀ m,n ∈ Natural

producto (cero, n) => 0
~~producto (Sucesor(m), n) => Sucesor (producto (m,n))~~
potencia (n, cero) => 1 => Sucesor(cero)
potencia (cero, n) => 0
potencia (n, Sucesor(m)) => Producto (n, potencia (n,m))

7.- En el .c de la carpeta.