___ METODOLOGÍA DE LA PROGRAMACIÓN (2º cuatrimestre). Curso 02-03
___ PROGRAMACION 1 (Plan viejo)

NOMBRE:____________________APELLIDOS:____

TIEMPO 2:00 h – Se pueden utilizar apuntes y libros

Los alumnos de Metodología de la Programación deben contestar obligatoriamente a las preguntas [3, 5 y 6] más 2 preguntas a elegir entre [1, 2 o 4]. Se pueden usar apuntes. Los alumnos de Programación 1 deben contestar las preguntas [1, 2, 3, 4, 5] Los alumnos de Laboratorio de programación deben contestar únicamente la pregunta [3]

1) Resuelve la siguiente ecuación de recurrencia. Los cálculos hay que presentarlos en hoja aparte. (2 ptos)

$$T(n) = \begin{cases} 1 & n = 0 \\ 2 & n = 1 \\ 2^n \cdot T(n-1)^4 & n \geq 2 \end{cases}$$

<u>Resultado</u>:

2) Considera el siguiente algoritmo y responde a las siguientes preguntas: (2 ptos)

```
Algoritmo Exa1
DATOS: A: vector[1..n] de {0,1,2,3} // Vector de números del 0 al 3
       B: vector[1..n+4] de N
RES:   R: N
METODO:
    R ← 0
    PARA i ← 1 HASTA n HACER
        j ← A[i]
        PARA K ← 1 HASTA j HACER
            R ← R + A[i]*B[i+k]
        FPARA
    FPARA
FExa1
```

a. ¿Cuál es la instrucción crítica de este algoritmo? ¿Cual es el tamaño (o talla) del algoritmo?

b. ¿Cuál es el caso mejor?. Calcula el coste para el caso mejor en función de la Instrucción crítica.

c. ¿Cuál es el caso peor?. Calcula el coste para el caso peor en función de la Instrucción crítica.

d. Suponiendo que cualquier vector sea equiprobable y que se pueden dar vectores con elementos repetidos. Calcula cual será el coste en el caso medio.

3) En la práctica 2 se analizaba la búsqueda binaria y algunas variaciones de la misma. Responde a las siguientes preguntas (2 ptos)

a. Indica si las siguientes expresiones son correctas, o no, para determinar la posición del pivote en el algoritmo de búsqueda binaria. (Rodea la respuesta correcta).

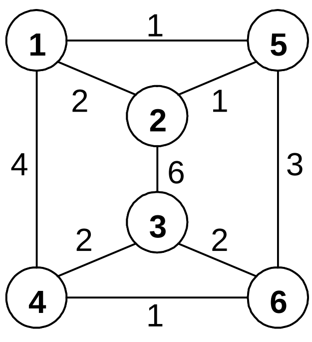
$$p = Ini + \left(\frac{Fin - Ini}{2}\right) \quad \text{V ó F} \quad ; \quad p = Fin - \left(\frac{Fin - Ini}{2}\right) \quad \text{V ó F}$$

b. En el caso del algoritmo de búsqueda ternaria, ¿podríamos utilizar las siguientes expresiones para calcular los dos pivotes centrales?

$$p_1 = Ini + \left(\frac{Fin - Ini}{3}\right) \text{ y } p_2 = Ini + 2 \cdot \left(\frac{Fin - Ini}{3}\right) \quad \text{Si ó No}$$

c. ¿Que algoritmo, el de búsqueda binaria o el de búsqueda ternaria, es mejor en comparaciones? ¿Son sus costes del mismo orden?. Indica de que orden es cada uno de ellos.

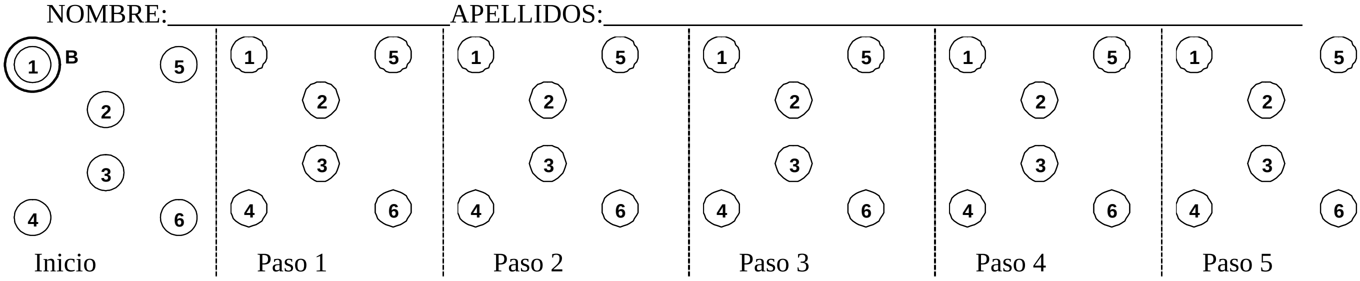
4) Considera el siguiente grafo conexo no dirigido con pesos y responde a las siguientes preguntas. y representa el proceso seguido por los algoritmos de Prim y Kruskal para obtener el camino básico considerando que en ambos casos se empieza por el nodo 1 (2 ptos)



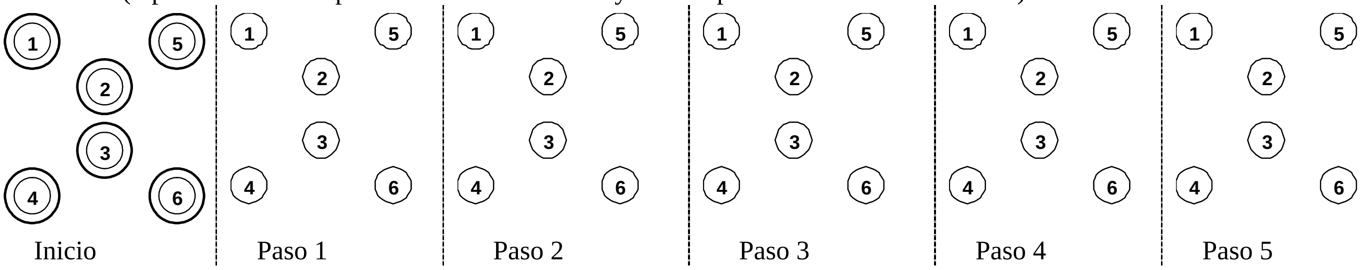
a. ¿En un grafo de N nodos completo, cuantos arcos tendrá el camino de extensión minimal? ¿Y si el grafo no es completo?

b. Realiza la traza para obtener el camino de extensión minimal utilizando el algoritmo de PRIM (utiliza los gráficos siguientes y representa en cada paso el arco seleccionado y el conjunto de nodos visitados y nodos no visitados).

NOMBRE:_____________________APELLIDOS:____



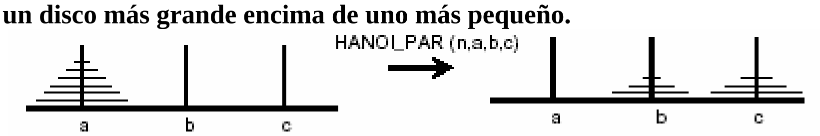
c. Realiza la traza para obtener el camino de extensión minimal utilizando el algoritmo de KRUSKAL (representa en cada paso el arco seleccionado y las componentes conexas existentes).



d. ¿Existe más de un camino de extensión minimal en el grafo del ejemplo? ¿Puede existir más de un camino de extensión minimal en un grafo cualquiera?. Justifica tu respuesta.

5) Utilizando como punto de partida el algoritmo de las Torres de Hanoi para n discos y 3 agujas. HANOI (n, a, b, c):

Se quiere implementar un algoritmo que en vez de situar todos los discos de la aguja A en la aguja C, sitúe los discos pares en la aguja C y los impares en la B, sabiendo que <u>el número de discos siempre va ha ser par</u> y que nunca puede existir un disco más grande encima de uno más pequeño.



Responde a las siguientes cuestiones en una hoja aparte (2 ptos):

a. Implementa el algoritmo. (Pista: Utiliza llamadas a la función HANOI normal para implementar el algoritmo. No es necesario implementar HANOI).

b. Escribe la ecuación de recurrencia del algoritmo y calcula su coste temporal en función del número de llamadas a la función MOV, si $T_{Hanoi}(n) = 2^n-1$

c. Calcula su coste espacial.

6) Pregunta de Fernando.(2 ptos)