

1 • ¿Cuál es la complejidad asintótica del algoritmo f si operar(n) es $O(n)$?

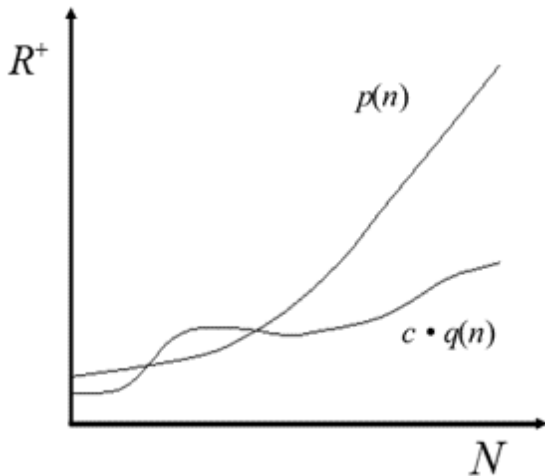
```
función f(n:natural):natural
  a,b:natural
  a <- 0
  b <- n
  mientras b >= 1 hacer
    a <- 2*operar(b)+a
    b <- b/2
  fmientras
  devolver a
ffunción
```

- ☐ $O(2*n)$
☐ $O(n^2)$
☒ $O(n \log n)$

2 • La complejidad asintótica del algoritmo de ordenación por selección es

- ☒ $O(n^2)$
☐ $O(n)$
☐ $O(n \log n)$

3 • Dada la siguiente gráfica donde p y q son funciones y c es una constante decimos que



- ☐ p(n) es de orden superior q(n)
☐ q(n) es de orden superior p(n)
☒ p(n) es de orden inferior q(n)

4 • De las siguientes etapas del diseño de algoritmos recursivos el orden de realización es

- ☐ Especificación, verificación y composición.
☐ Análisis de casos, verificación y especificación.

5 ☒ **El siguiente algoritmo ¿está bien definido?**

```
función obtener(x:natural, y:natural):natural
  si x =1
    devolver x+y
  si no
    devolver obtener(x-1,y+1)
  fsi
ffunción
```

- ☒ Sí.
- ☐ No, porque no está definido para los casos en que $x \leq 0$.
- ☐ No, porque no hay un caso base para comprobar el valor de y .

6 ☒ **En el siguiente algoritmo**

```
función f(n:real):real
  x:real
  si n <= 2
    devolver n
  si no
    x <- f(n-1) * n
    devolver x
  fsi
ffunción
```

- ☐ Podemos obtener la complejidad asintótica aplicando la descomposición por niveles.
- ☒ Podemos obtener la complejidad asintótica aplicando la fórmula correspondiente a la disminución del problema por sustracción.
- ☐ Ambas respuestas son correctas.

7 ☒ **Para convertir un algoritmo recursivo lineal final en iterativo**

- ☒ Es suficiente con utilizar un bucle que lleve el parámetro al caso base.
- ☐ Es necesario realizar dos bucles, uno que lleve el parámetro al caso base y otro que vaya recomponiendo la solución.
- ☐ En ocasiones, solamente se puede realizar mediante una pila para ir guardando los valores.

8 ☒ **¿Qué tipo de recursividad presenta el algoritmo calcular cuya definición en el caso general es $\text{calcular}(n) = \text{calcular}(g(n-2))$, y $g(n) = n/2$?**

- ☒ Lineal final.
- ☐ Lineal no final.
- ☐ Anidada

9 ● El algoritmo del Torneo que resuelve la distribución de parejas de un torneo de pádel presenta una recursividad

- ☐ lineal final.
- ☒ lineal no final.
- ☐ múltiple.

10 ● La función combinar(V,izq,centro,der) del algoritmo quicksort

- ☐ ordena el vector V desde la posición izq a la posición der
- ☐ tiene una complejidad del $O(n \log n)$, siendo n el tamaño del vector
- ☒ Ninguna respuesta es correcta.

11 ● La complejidad temporal del algoritmo mergesort para un determinado tamaño del problema n

- ☐ depende del contenido de los datos.
- ☐ es del $O(\log n)$.
- ☒ Ninguna respuesta es correcta.

12 ● ¿Cuál de estas afirmaciones es falsa?

- ☐ El algoritmo mergesort utiliza una estrategia divide y vencerás.
- ☒ El algoritmo quicksort tiene una complejidad asintótica menor que el algoritmo mergesort.
- ☐ El algoritmo quicksort puede llegar a tener una complejidad asintótica del $O(n^2)$.

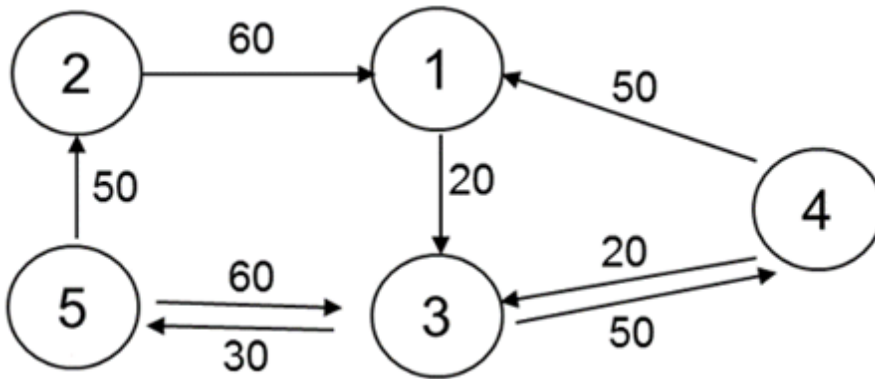
13 ● El algoritmo de Dijkstra calcula el coste del camino más corto

- ☐ entre dos vértices de un grafo.
- ☐ entre todos los vértices de un grafo.
- ☒ Ninguna respuesta es correcta.

14 ● Un técnico informático tiene al principio del día una lista de peticiones de clientes correspondientes a averías que debe atender y quiere minimizar el tiempo total de atención de todos los clientes, considerando como tiempo de atención para un cliente la suma del tiempo de reparación de la avería de cada cliente y el tiempo de espera del cliente hasta que se atiende su petición desde el principio de la mañana. La forma más eficaz de resolver este algoritmo es utilizando

- ☐ Un algoritmo voraz, ordenando los procesos en orden ascendente por el tiempo de atención.
- ☐ Un algoritmo voraz, ordenando los procesos en orden ascendente por el tiempo de espera.
- ☒ Ninguna respuesta es correcta.

15 ● ¿Se puede aplicar el algoritmo de Prim al siguiente grafo?



- ☒ Sí.
- ☐ No, porque el grafo es dirigido.
- ☐ No, porque hay vértices que no tienen arista entre ellos.

16 ● Supongamos que tenemos un grafo que representa las poblaciones de la Comunidad Valenciana de más de 100000 habitantes junto con todas las carreteras de conexión entre ellas. Se quiere obtener un recorrido que pase por todas estas ciudades una única vez y volver a la ciudad de origen de forma que se recorra el menor número de kilómetros. Si aplicamos una estrategia voraz para resolver este problema obtendremos

- ☐ La solución óptima.
- ☐ Puede que no encuentre ninguna solución aunque ésta exista.
- ☒ Una solución factible.

17 ● El algoritmo de Floyd estudiado en la asignatura utiliza una metodología del tipo

- ☒ programación dinámica.
- ☐ voraz.
- ☐ divide y vencerás.

18 ● La estrategia de programación más eficiente para resolver el problema de la mochila cuando los objetos no se pueden fraccionar, los pesos son enteros y los valores reales es

- ☒ programación dinámica.
- ☐ voraz.
- ☐ ramificación y poda.

19 ● El número de combinaciones de n elementos cogidos de k en k se puede resolver aplicando

- ☒ Algoritmos recursivos.
- ☐ Algoritmos voraces.
- ☐ Ninguna respuesta es correcta.

20 ● Dada la siguiente matriz resultante de aplicar el algoritmo de Floyd correspondiente al vértice 3, donde INF indica un valor muy alto, ¿cuál es el

resultado de la matriz tras aplicarlo para el vértice 4?

0	10	14	5	34
10	0	12	10	24
30	32	0	20	20
23	10	20	0	3
10	20	INF	INF	0

- ☐ Fila 1: 0, 10, 14, 5, 8 ; Fila 2: 10, 0, 12, 10, 13 ; Fila 3: 30, 32, 0, 20, 20 ; Fila 4: 23,10, 20, 0, 3 ; Fila 5: 10, 20, INF, INF, 0
- ☒ Fila 1: 0, 10, 14, 5, 8 ; Fila 2: 10, 0, 12, 10, 13; Fila 3: 30, 30, 0, 20, 20 ; Fila 4: 23,10, 20, 0, 3 ; Fila 5: 10, 20, INF, INF, 0
- ☐ Ninguna respuesta es correcta.

21 ☒ **En los algoritmos de vuelta atrás, cuando no hay más candidatos en un nivel**

- ☐ Se ha llegado al final.
- ☐ Se continúa buscando en el siguiente nivel.
- ☒ Se continúa buscando en el nivel anterior.

22 ☒ **En el problema de la mochila con vuelta atrás para una mochila de objetos con pesos 30, 40, 35, 20 y valores 35, 30 y 20, 40 con capacidad máxima de 60, suponiendo que la solución óptima hasta ese momento es 70, ¿cuántos nodos no son factibles en la rama que se expande a partir del nodo (1,0,__,_) ?**

- ☐ 0
- ☐ 1
- ☒ 2

23 ☒ **En la transformación a iterativo de los algoritmos de vuelta atrás recursivos**

- ☐ La función SiguienteCandidatoNivel se sustituye por incrementar en una unidad el valor de la variable que indica el nivel del árbol (variable k).
- ☒ La función ExisteCandidatoNivel se incluye dentro de una función si en vez de un mientras.
- ☐ La llamada recursiva se sustituye por disminuir en una unidad el valor de la variable que indica el nivel del árbol (variable k).

24 ☒ **El problema del viajante de comercio resuelto por vuelta atrás tiene una complejidad**

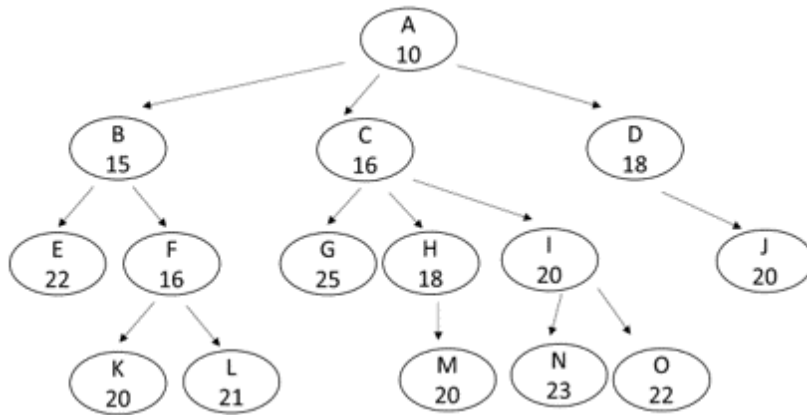
- ☐ Exponencial.
- ☒ Factorial.
- ☐ Polinómica.

25 ☒ **¿Cuál de las siguientes afirmaciones es falsa?**

- ☐ Los algoritmos voraces suelen tener una complejidad menor que los de vuelta atrás.
- ☒ Los algoritmos de ramificación y poda suelen tener una complejidad asintótica menor que los algoritmos voraces.
- ☐ Los algoritmos de programación dinámica suelen tener una complejidad menor que los algoritmos de ramificación y poda.

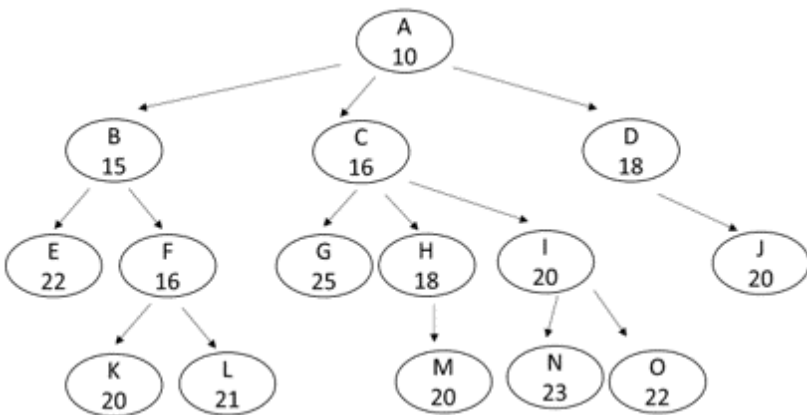
26 ☒ **Supongamos el siguiente árbol de expansión de un problema de**

ramificación y poda cuyo objetivo es minimizar una función, ¿en qué orden se seleccionan los nodos siguiendo una estrategia MC-FIFO?



- ☒ A, B, C, F, D, H
☐ A, B, F, C, H, D
☐ Ninguna respuesta es correcta.

27 ● Supongamos el siguiente árbol de expansión de un problema de ramificación y poda cuyo objetivo es minimizar una función, ¿cuál es el orden de generación de nodos siguiendo una estrategia MC-LIFO?



- ☐ A, B, C, D, E, F, G, H, I, K, L, D, J, M
☐ A, B, C, D, E, F, K, L, G, H, I, M, J
☒ Ninguna respuesta es correcta.

28 ● ¿Qué hace la función Admisible en el algoritmo que resuelve el problema de la Mochila?

- ☒ Poda los nodos cuya cota es menor que el valor de la solución actual.
☐ Poda los nodos cuyo valor es menor que el valor de la solución actual.
☐ Poda los nodos cuyo peso es mayor que el peso de la mochila.

29 ● ¿Cuál es la función $T(n)$ del siguiente algoritmo?
 (NOTA: La expresión $\sum_{j=1}^x \{...\}$ indica el sumatorio desde $j=1$ hasta x de todas las instrucciones que se incluyan en los corchetes.)

```

función f(V:entero[n]):entero
    a,i,z:entero
(1)    a <- 1
(2)    z <- 0
(3)    mientras a <= n-1 hacer
(4)        para i<-1 hasta a hacer
(5)            z <- z + Va - i
(6)        fpara
(7)            a <- a + 1
(8)    fmientras
(9)    devolver z
ffunción

```

- ☒ $T(n) = t(1) + t(2) + t(3) + \sum_{a=1}^{n-1} [t(4) + a * t(5) + t(7)] + t(9)$
- ☐ $T(n) = t(1) + t(2) + t(3) + \sum_{i=1}^a [t(4) + a * t(5) + t(7)] + t(9)$
- ☐ Ninguna respuesta es correcta.

30 ☒ ¿Cuál de las siguientes expresiones es falsa?

- ☒ $O(n) \subset O(n^n) \subset O(n!)$
- ☐ $O(n) \subset O(n \log n) \subset O(n!)$
- ☐ $O(n \log n) \subset O(n^2) \subset O(2^n)$

Imprimir esta página