Se pretende implementar mediante programación dinámica iterativa la función recursiva:

```
int f( int x, int y ) {
  if( x <= y ) return 1;
  return x + f(x-1,y);
}
```

¿Cuál es la mejor complejidad espacial que se puede conseguir?

Seleccione una:

- a. $O(x^2)$
- b. $O(x)$
- c. $O(1)$ ✓

Dada la suma de la recurrencia

$$T(n) = \begin{cases} 1 & n = 0 \\ \sum_{k=0}^{n-1} T(k) & n > 0 \end{cases}$$

¿cuál de las siguientes afirmaciones es cierta?

Seleccione una:

- a. $T(n) \in \Theta(2^n)$ ✓
- b. $T(n) \in \Theta(n^2)$
- c. $T(n) \in \Theta(n!)$

¿Para qué se utiliza el TAD "Union-find" en el algoritmo de Kruskal?

Seleccione una:

- a. Para comprobar si un arco forma ciclos.
- b. Para comprobar si un vétice ya ha sido visitado.
- c. Para comprobar si dos vértices son equivalentes.

Se pretende implementar mediante programaciÛn din·mica iterativa la funciÛn recursiva:

```
unsigned f( unsigned y, unsigned x) // suponemos y >= x
if (x==0 || y==x) return 1;
return f(y-1, x-1) + f(y-1, x);
```

øCu·l es la mejor complejidad espacial que se puede conseguir?

{
~O(y^2)
=O(y)
~O(1)
}

¿Cuál es la complejidad temporal de la siguiente función?

```
int f(int n){
    int k=0;
    for (int i = n; i > 0; i/=3)
        for (int j=i; j > 0; j-=3)
            k++;
    return k;
}
```

Seleccione una:

- a. $\Theta(n^2)$
- b. $\Theta(n\log n)$
- c. $\Theta(n)$

Se desea encontrar el camino mas corto entre dos ciudades. Para ello se dispone de una tabla con la distancia entre los pares de ciudades en los que hay carreteras o un valor centinela (por ejemplo, -1) si no hay, por lo que para ir de la ciudad inicial a la final es posible que haya que pasar por varias ciudades. También se conocen las coordenadas geográficas de cada ciudad y por tanto la distancia geométrica (en línea recta) entre cada par de ciudades. Se pretende acelerar la búsqueda de un algoritmo de ramificación y poda priorizando los nodos vivos (ciudades) que estén a menor distancia geográfica de la ciudad objetivo.

(a) El nuevo algoritmo no garantiza que vaya a ser más rápido para todas las instancias del problema posibles.

(b) El nuevo algoritmo siempre será más rápido.

(c) Esta estrategia no asegura que se obtenga el camino mas corto.

10. La mejor solución que se conoce para el problema de la mochila continua sigue el esquema…

    (a) …divide y vencerás.
    (b) …ramificación y poda.
    (c) ...voraz. En Voraz no es Óptima?

19. Decid cuál de estas tres es la cota pesimista más ajustada al valor óptimo de la mochila discreta: Diferencias entre continua y discreta

    (a) El valor de la mochila continua correspondiente
    (b) El valor de una mochila que contiene todos los objetos aunque se pase del peso máximo permitido
    (c) El valor de la mochila discreta que se obtiene usando un algoritmo voraz basado en el valor específico de los objetos

20. Un problema de tamaño n puede transformarse en tiempo $O(n^2)$ en otro de tamaño n - 1. Por otro lado, la solución al problema cuando la talla es 1 requiere un tiempo constante. ¿cuál de estas clases de coste temporal asintótico es la más ajustada?

(a) $O(2^n)$
(b) $O(n^3)$

(c) $O(n^2)$

23. Sea A una matriz cuadrada n x n. Se trata de buscar una permutación de las columnas tal que la suma de los elementos de la diagonal de la matriz resultante sea mínima. Indicad cuál de las siguientes afirmaciones es falsa.

(a) La complejidad temporal de la mejor solución posible problema es O(n!).

(b) Si se construye una solución al problema basada en el esquema de ramificación y poda, una buena elección de cotas optimistas y pesimistas podría evitar la exploración de todas las permutaciones posibles.

(c) La complejidad temporal de la mejor solución posible al problema es $O(n^2)$.

24. Cuál de los siguientes algoritmos proveería una cota pesimista para el problema de encontrar el camino mas corto entre dos ciudades (se supone que el grafo es conexo).

    (a) Calcular la distancia geométrica (en línea recta) entre la ciudad origen y destino.

    (b) Para todas las ciudades que son alcanzables en un paso desde la ciudad inicial, sumar la distancia a dicha ciudad y la distancia geométrica hasta la ciudad destino.

    (c) Calcular la distancia recorrida moviéndose al azar por el grafo hasta llegar (por azar) a la ciudad destino.

26. Dadas las siguientes funciones:

```
// Precondición: { 0 <= i < v.size(); i < j <= v.size() }
unsigned f( const vector<unsigned>&v, unsigned i, unsigned j ) {
  if( i == j+1 )
    return v[i];
  unsigned sum = 0;
  for( unsigned k = 0; k < j - i; k++ )
    sum += f( v, i, i+k+1) + f( v, i+k+1, j );
  return sum;
}

unsigned g( const vector<unsigned>&v ) {
  return f( v, v.begin(), v.end() );
}
```

Se quiere reducir la complejidad temporal de la función `g` usando programación dinámica iterativa. ¿cuál sería la complejidad espacial?

(a) cúbica

[(b)] cuadrática

(c) exponencial

28. En los algoritmos de ramificación y poda, ¿el valor de una cota pesimista es mayor que el valor de una cota optimista? (se entiende que ambas cotas se aplican sobre el mismo nodo)

    (a) No, nunca es así.

    (b) En general sí, si se trata de un problema de minimización, aunque en ocasiones ambos valores pueden coincidir.

    (c) En general sí, si se trata de un problema de maximización, aunque en ocasiones ambos valores pueden coincidir.

29. Cuando se resuelve el problema de la mochila discreta usando la estrategia de vuelta atrás, ¿puede ocurrir que se tarde menos en encontrar la solución óptima si se prueba primero a meter cada objeto antes de no meterlo?

    (a) Sí, tanto si se usan cotas optimistas para podar el árbol de búsqueda como si no.

    (b) No, ya que en cualquier caso se deben explorar todas las soluciones factibles.

    (c) Sí, pero sólo si se usan cotas optimistas para podar el árbol de búsqueda.

31. En el esquema de vuelta atrás el orden en el que se van asignando los distintos valores a las componentes del vector que contendrá la solución...

8

(a) … es irrelevante si no se utilizan mecanismos de poda basados en la mejor solución hasta el momento.

(b) … puede ser relevante si se utilizan mecanismos de poda basados en estimaciones optimistas.

(c) Las otras dos opciones son ciertas.

37. Se quieren ordenar d números distintos comprendidos entre 1 y n. Para ello se usa un array de n booleanos que se inicializan primero a false. A continuación se recorren los d números cambiando los valores del elemento del vector de booleanos correspondiente a su numero a true. Por último se recorre el vector de booleanos escribiendo los índices de los elementos del vector de booleanos que son true. ¿Es este algoritmo más rápido (asintóticarnente) que el mergesort?

(a) Sí, ya que el mergesort es log n) y este es O(n)

(b) Sólo si d log d > (k * n) (donde k es una constante que depende de la implementación)

(c) No, ya que este algoritmo ha de recorrer varias veces el vector de booleanos.

13. Si $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$ entonces ...

(a) $\ldots f(n) \in \Theta(g(n))$

(b) $\ldots f(n) \in O(g(n))$

(c) $\ldots g(n) \in O(f(n))$

18. Sea la siguiente relación de recurrencia

$$T(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ 2T(\frac{n}{2}) + g(n) & \text{en otro caso} \end{cases}$$

Si $T(n) \in O(n)$, ¿en cuál de estos tres casos nos podemos encontrar?

(a) $g(n) = n^2$

(b) $g(n) = n$ ¿No se hace con la fórmula?

(c) $g(n) = 1$

19. Los algoritmos de vuelta atrás que hacen uso de cotas optimistas generan las soluciones posibles al problema median

(a) . . . un recorrido guiado por una cola de prioridad de donde se extraen primero los nodos que representan los subárboles más prometedores del espacio de soluciones.

(b) . . . un recorrido guiado por estimaciones de las mejores ramas del árbol que representa el espacio de soluciones.

(c) . . . un recorrido en profundidad del árbol que representa el espacio de soluciones.

32. La siguiente relación de recurrencia expresa la complejidad de un algoritmo recursivo, donde g(n) es una función polinómica:

$$T(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ 2T(\frac{n}{2}) + g(n) & \text{en otro caso} \end{cases}$$

Di cuál de las siguientes afirmaciones es falsa:

(a) Si $g(n) \in O(n^2)$ la relación de recurrencia representa la complejidad temporal del algoritmo de búsqueda por inserción.

(b) Si $g(n) \in O(n)$ la relación de recurrencia representa la complejidad temporal del algoritmo de ordenación *mergesort*.

(c) Si $g(n) \in O(1)$ la relación de recurrencia representa la complejidad temporal del algoritmo de búsqueda dicotómica.

39. Sea A una matriz cuadrada $n \times n$. Se trata de buscar una permutación de las columnas tal que la suma de los elementos de la diagonal de la matriz resultante sea mínima. Indicad cuál de las siguientes afirmaciones es falsa.

(a) Si se construye una solución al problema basada en el esquema de ramificación y poda, una buena elección de cotas optimistas y pesimistas podría evitar la exploración de todas las permutaciones posibles.

(b) La complejidad temporal de la mejor solución posible al problema es $O(n \log n)$.

(c) La complejidad temporal de la mejor solución posible al problema está en $\Omega(n^2)$.

18. Si $\lim_{n\to\infty}(f(n)/n^2) = k$, y $k \neq 0$, ¿cuál de estas tres afirmaciones es *falsa*?

(a) $f(n) \in O(n^3)$

[(b)] $f(n) \in \Theta(n^3)$

(c) $f(n) \in \Theta(n^2)$

17. La función γ de un número semientero positivo (un número es semientero si al restarle 0.5 es entero) se define como:

```
double gamma( double n ) { // Se asume n>=0.5 y n-0.5 entero
  if( n == 0.5 )
    return sqrt(PI);
  return n * gamma( n - 1);
}
```

¿Se puede calcular usando programación dinámica iterativa?

(a) No, ya que el índice del almacén sería un número real y no entero.

(b) No, ya que no podríamos almacenar los resultados intermedios en el almacén.

[(c)] Sí, pero la complejidad temporal no mejora.

21. ¿Cuál de estas afirmaciones es *falsa*?

(a) La solución de programación dinámica iterativa al problema de la mochila discreta realiza cálculos innecesarios.

(b) Los algoritmos iterativos de programación dinámica utilizan memoización para evitar resolver de nuevo los mismos subproblemas que se vuelven a presentar.

(c) La memoización evita que un algoritmo recursivo ingenuo resuelva repetidamente el mismo problema.

10. En el problema del coloreado de grafos (mínimo número de colores necesarios para colorear todos los vértices de un grafo de manera que no queden dos adyacentes con el mismo color) resuelto mediante *ramificación y poda*, una cota optimista es el resultado de asumir que ...

(a) ...se van a utilizar tantos colores distintos a los ya utilizados como vértices quedan por colorear.

(b) ...no se van a utilizar colores distintos a los ya utilizados.

(c) ...sólo va a ser necesario un color más.

13. Se quiere reducir la complejidad temporal de la siguiente función haciendo uso de programación dinámica. ¿Cuál sería la complejidad temporal resultante?

```
unsigned g( unsigned n, unsigned r){
   if (r==0 || r==n)
      return 1;
   return g(n-1, r-1) + g(n-1, r);
}
```

[(a)] Cuadrática
(b) Se puede reducir hasta lineal.
(c) La función no cumple con los requisitos necesarios para poder aplicar programación dinámica.

22. Si $\lim_{n\to\infty}(f(n)/n^2) = k$, y $k \neq 0$, ¿cuál de estas tres afirmaciones es *cierta*?

(a) $f(n) \in \Omega(n^3)$
[(b)] $f(n) \in \Theta(n^2)$
(c) $f(n) \in \Theta(n^3)$

24. Si $f \in \Theta(g_1)$ y $f \in \Theta(g_2)$ entonces

(a) $f \in \Theta(g_1 \cdot g_2)$
(b) $f \notin \Theta(\text{máx}(g_1, g_2))$
[(c)] $f \in \Theta(g_1 + g_2)$

25. ¿Cuál es la complejidad temporal de la siguiente función?

```
unsigned examen (unsigned n) {
unsigned i=n, k=0;
   while (i>0){
       unsigned j=i;
       do{
          j = j * 2;
          k = k + 1;
       } while (j<=n);
       i = i / 2;
   }
   return k;
}
```

[(a)] $\Theta(\log^2 n)$ ¿Por que log?

(b) $\Theta(\log n)$

(c) $\Theta(n)$

3. Decid cuál de estas tres es la cota optimista más ajustada al valor óptimo de la mochila discreta:

   (a) El valor de una mochila que contiene todos los objetos aunque se pase del peso máximo permitido.

   [(b)] El valor de la mochila continua correspondiente .

   (c) El valor de la mochila discreta que se obtiene usando un algoritmo voraz basado en el valor específico de los objetos.

5. ¿En ramificación y poda, tiene sentido utilizar la cota optimista de los nodos como criterio para ordenar la lista de nodos vivos?

   (a) Sí, en el caso de que se ordene la lista de nodos vivos, siempre debe hacerse según el criterio de la cota optimista.

   (b) No, la cota optimista sólo se utiliza para determinar si una *n-tupla* es prometedora.

   [(c)] Sí, aunque no es una garantía de que sea una buena estrategia de búsqueda.

6. ¿Qué estrategia de búsqueda es a priori más apropiada en un esquema de vuelta atrás?

   (a) Explorar primero los nodos que están más completados.

   [(b)] En el esquema de vuelta atrás no se pueden definir estrategias de búsqueda.

   (c) Explorar primero los nodos con mejor cota optimista.

7. Sea la siguiente relación de recurrencia

$$T(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ 2T(\frac{n}{2}) + g(n) & \text{en otro caso} \end{cases}$$

   Si $T(n) \in O(n)$, ¿en cuál de estos tres casos nos podemos encontrar?

   [(a)] Las otras dos opciones son ambas ciertas.

   (b) $g(n) = \log n$

   (c) $g(n) = \sqrt{n}$

12. ¿Qué nos proporciona la media entre el coste temporal asintótico (o complejidad temporal) en el peor caso y el coste temporal asintótico en el mejor caso?

[(a)] Nada de interés.
(b) El coste temporal promedio.
(c) El coste temporal asintótico en el caso medio.

19. ¿Cuál sería la complejidad temporal de la siguiente función tras aplicar programación dinámica?

```
double f(int n, int m) {
   if(n == 0) return 1;
   return m * f(n-1,m) * f(n-2,m);
}
```

[(a)] $\Theta(n)$
(b) $\Theta(n^2)$
(c) $\Theta(n \times m)$

22. Se desea resolver el problema de la potencia enésima (x^n), asumiendo que n es par y que se utilizará la siguiente recurrencia: `pot(x,n) = pot(x,n/2) * pot(x,n/2);` ¿Qué esquema resulta ser más eficiente en cuanto al coste temporal?

(a) En este caso tanto programación dinámica como divide y vencerás resultan ser equivalentes en cuanto a la complejidad temporal.

[(b)] Programación dinámica.

(c) Divide y vencerás.

26. ¿Qué se deduce de $f(n)$ y $g(n)$ si se cumple $\lim_{n\to\infty}(f(n)/g(n)) = k$, con $k \neq 0$?

[(a)] $f(n) \in O(g(n))$ y $g(n) \in O(f(n))$

(b) $f(n) \in O(g(n))$ pero $g(n) \notin O(f(n))$

(c) $g(n) \in O(f(n))$ pero $f(n) \notin O(g(n))$

29. Un tubo de n centímetros de largo se puede cortar en segmentos de 1 centímetro, 2 centímetros, etc. Existe una lista de los precios a los que se venden los segmentos de cada longitud. Una de las maneras de cortar el tubo es la que más ingresos nos producirá. Se quiere resolver el problema mediante vuelta atrás. ¿cual sería la forma más adecuada de representar las posibles soluciones?

(a) una tabla que indique, para cada posición donde se va a cortar, cada uno de los posibles valores acumulados.

[(b)] un vector de booleanos.

(c) un par de enteros que indiquen los cortes realizados y el valor acumulado.

33. Dado un problema de maximización resuelto mediante un esquema de ramificación y poda, ¿qué ocurre si la cota optimista resulta ser un valor excesivamente elevado?

(a) Que se podría explorar menos nodos de los necesarios.

(b) Que se podría podar el nodo que conduce a la solución óptima.

[(c)] Que se podría explorar más nodos de los necesarios.

39. Dado el problema del laberinto con tres movimientos, ¿cuál de las estrategias siguientes proveería de una cota optimista para ramificación y poda?

(a) Suponer que ya no se van a realizar más movimientos.

[(b)] Las otras dos estrategias son ambas válidas.

(c) Suponer que en adelante todas las casillas del laberinto son accesibles.