| | |
|---|---|
| **Comenzado el** | jueves, 19 de junio de 2025, 17:00 |
| **Estado** | Finalizado |
| **Finalizado en** | jueves, 19 de junio de 2025, 17:16 |
| **Tiempo empleado** | 15 minutos 56 segundos |
| **Calificación** | **3,67** de 10,00 (**36,67**%) |

**Pregunta 1**

Sin contestar

Se puntúa como 0 sobre 1,00

Dado el siguiente bucle con su especificación:

$P = \{0 \leq N \leq w.size\,(\,)\,\}$

```
int c = 0, a = N-1;
while (0 <= a){
  if(w[a] % 2 == 0)
       c = c + 1;
  a = a - 1;
}
```

$Q = \{c = \#i{:}\, 0 \leq i < N{:}\, w[i]\,\%\,2 = 0\}$

¿Cuál de estas propiedades es invariante?

Seleccione una:

- a. $c = \#i{:}\, a < i \leq N{:}\, w[i]\,\%\,2 = 0.$
- b. $0 \leq a < v.size\,(\,)$
- c. $c = \#i{:}\, a < i < N{:}\, w[i]\,\%\,2 = 0.$
- d. Ninguna de las anteriores.

a. Falso. Incorrecta. La expresión es incorrecta para el valor $i = N$ ya que $N$ puede ser igual a $v.size\,(\,)$.
b. Falso. Incorrecta. El invariante debe ser cierto al terminar el bucle y en ese momento $a = -1$
c. Cierto. El bucle cuenta el número de valores que son pares. La variable $a$ comienza valiendo $N - 1$ y termina valiendo $-1$. En cada iteración, si el valor del vector es par, se aumenta en uno la variable $c$, y se disminuye en uno la variable del bucle $a$. En una iteración cualquiera, la variable $c$ acumula el número de valores pares desde la posición $a + 1$ hasta la $N - 1$ ($\#i{:}\, a < i < N{:}\, w[i]\,\%\,2 = 0$).
d. Falso. La respuesta correcta es: $c = \#i{:}\, a < i < N{:}\, w[i]\,\%\,2 = 0.$

La respuesta correcta es: $c = \#i{:}\, a < i < N{:}\, w[i]\,\%\,2 = 0.$

**Pregunta 2**

Sin contestar

Se puntúa como 0 sobre 1,00

Dados los algoritmos de Mergesort (ordenación por mezclas) y Quicksort (ordenación rápida) en el que se elige como pivote el primer elemento del segmento, ¿Cuál de las siguientes afirmaciones es correcta siendo $n$ el tamaño de entrada del vector?.

Seleccione una:

- a. El coste en el caso peor de Mergesort y el coste en el caso peor de Quicksort coincide y es $\Theta\left(n^2\right)$.
- b. El coste en el caso peor de Mergesort y el coste en promedio de Quicksort coincide y es $\Theta\left(n\,\mathbf{log}\;n\right)$.
- c. El coste en promedio de Mergesort coincide con el coste en promedio de Quicksort y es $\Theta\left(\mathbf{log}\;n\right)$.
- d. Ninguna de las anteriores.

a. Falso. Incorrecta. Mergesort siempre hace dos llamadas recursivas con la mitad de tamaño, incluído en su caso peor, por lo que su coste siempre es $M(n) = c$ si $n \leq 1$, $M(n) = 2M\left(n/2\right) + c'n$ si $n > 1$, de donde $M(n) \in \Theta\left(n\,\mathbf{log}\;n\right)$. En cambio Quicksort, en el caso peor, hace una llamada recursiva con tamaño $n-1$ y otra vacía, obteniendo la siguiente recurrencia del coste: $Q(n) = c$ si $n \leq 1$, $Q(n) = Q(n-1) + c'n$ si $n > 1$, de donde $Q(n) \in \Theta\left(n^2\right)$. Que no coinciden.
b. Cierto. Mergesort siempre hace dos llamadas recursivas con la mitad de tamaño, mientras que en promedio Quicksort hace también dos llamadas con la mitad de tamaño. En estos casos la recurrencia del coste es $T(n) = c$ si $n \leq 1$, $T(n) = 2T\left(n/2\right) + c'n$ si $n > 1$, de donde $T(n) \in \Theta\left(n\,\mathbf{log}\;n\right)$.
c. Falso. Incorrecta. Mergesort siempre hace dos llamadas recursivas con la mitad de tamaño, incluído en promedio. Igualmente Quicksort, en promedio, también hace dos llamadas con la mitad de tamaño. En ambos casos la recurrencia del coste es $T(n) = c$ si $n \leq 1$, $T(n) = 2T\left(n/2\right) + c'n$ si $n > 1$, de donde $T(n) \in \Theta\left(n\,\mathbf{log}\;n\right)$.
d. Falso. La respuesta correcta es: El coste en el caso peor de Mergesort y el coste en promedio de Quicksort coincide y es $\Theta\left(n\,\mathbf{log}\;n\right)$.

La respuesta correcta es: El coste en el caso peor de Mergesort y el coste en promedio de Quicksort coincide y es $\Theta\left(n\,\mathbf{log}\;n\right)$.

**Pregunta 3**

Correcta

Se puntúa 1,00 sobre 1,00

Indica cuál de las siguientes es una función de cota que permite demostrar la terminación de este bucle, suponiendo que $0 \leq v.size()$:

```
int i = 0, j = 1;
while (j <= v.size()){
  if (v[i] > v[j])
       i = j;
  j = j + 1;
}
```

Seleccione una:

- a. $v.size() - j - 3$.
- b. $j$.
- c. $v.size() - j + 1$ . Cierto. La variable $j$ cumple $1 \leq j \leq v.size() + 1$ así que cuando $j \leq v.size()$ se cumple que $v.size() + 1 - j \geq 0$. Además, puesto que $j$ se incrementa uno a uno, $v.size() + 1 - j$ decrece en cada vuelta.
- d. Ninguna de las anteriores.

a. Falso. Incorrecta. Por ejemplo, cuando $j = v.size()$, $v.size() - j - 3 < 0$
b. Falso. Incorrecta. La variable $j$ no decrece.
c. Cierto. La variable $j$ cumple $1 \leq j \leq v.size() + 1$ así que cuando $j \leq v.size()$ se cumple que $v.size() + 1 - j \geq 0$. Además, puesto que $j$ se incrementa uno a uno, $v.size() + 1 - j$ decrece en cada vuelta.
d. Falso. La respuesta correcta es: $v.size() - j + 1$.

La respuesta correcta es: $v.size() - j + 1$.

**Pregunta 4**

Sin contestar

Se puntúa como 0 sobre 1,00

Indica cuál de las siguientes afirmaciones es incorrecta

Seleccione una:

- a. $O(\sqrt{n}) \subset O(n^3)$
- b. $\Theta(n^2) = \Theta(4n^2)$
- c. $O(2^n) = O(3^n)$
- d. Ninguna de las anteriores.

a. Falso. Afirmación correcta. Por el teorema del límite, como $\lim_{n \to \infty} \frac{\sqrt{n}}{n^3} = 0$, $O(\sqrt{n}) \subset O(n^3)$.
b. Falso. Afirmación correcta. Por el teorema del límite, como $\lim_{n \to \infty} \frac{4n^2}{n^2} = 4$, $4n^2 \in \Theta(n^2)$.
c. Cierto. Por el teorema del límite, como $\lim_{n \to \infty} \frac{2^n}{3^n} = 0$, $O(3^n) \not\subseteq O(2^n)$.
d. Falso. La respuesta correcta es: $O(2^n) = O(3^n)$

La respuesta correcta es: $O(2^n) = O(3^n)$

**Pregunta 5**

Sin contestar

Se puntúa como 0 sobre 1,00

El coste $T(n)$ de un algoritmo recursivo viene dado por la siguiente recurrencia. ¿A qué $\Theta$ pertenece dicho coste?

$T(n) = 1$ si $n \leq 2$

$T(n) = T(n-3) + 2n$ si $n > 2$

Seleccione una:

- a. $\Theta(2^n)$
- b. $\Theta(n^2)$
- c. $\Theta(log\, n)$
- d. Ninguna de las anteriores.

a. Falso. Aplicando el teorema de la resta con a=1, b=3 y k=1, como $a = 1$ tenemos que $T(n) \in \Theta(n^2) \neq \Theta(2^n)$.
b. Cierto. Aplicando el teorema de la resta con a=1, b=3 y k=1, como $a = 1$ tenemos que $T(n) \in \Theta\left(n^{k+1}\right) = \Theta(n^2)$.
c. Falso. Aplicando el teorema de la resta con a=1, b=3 y k=1, como $a = 1$ tenemos que $T(n) \in \Theta(n^2) \neq \Theta(\mathbf{log}\, n)$.
d. Falso. La respuesta correcta es: $\Theta(n^2)$

La respuesta correcta es: $\Theta(n^2)$

**Pregunta 6**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuál de las siguientes condiciones es un requisito imprescindible para realizar el algoritmo de búsqueda binaria sobre un vector?

Seleccione una:

- a. El vector debe ser de valores enteros.
- b. No hay ningún requisito imprescindible: se puede aplicar sobre cualquier vector.
- c. Los valores deben estar ordenados. ✔ Cierto. Los valores del vector deben estar ordenados (ya sea creciente o decrecientemente) según un orden bien definido.
- d. Ninguna de las anteriores.

a. Falso. Incorrecta. El algoritmo de búsqueda binaria se puede aplicar sobre vectores de cualquier tipo, siempre y cuando estos elementos estén ordenados. Por ejemplo: char, string, float...
b. Falso. Incorrecta. Se puede aplicar sobre vectores cuyos valores estén ordenados (ya sea creciente o decrecientemente) según un orden bien definido.
c. Cierto. Los valores del vector deben estar ordenados (ya sea creciente o decrecientemente) según un orden bien definido.
d. Falso. La respuesta correcta es: Los valores deben estar ordenados.

La respuesta correcta es: Los valores deben estar ordenados.

**Pregunta 7**

Correcta

Se puntúa 1,00 sobre 1,00

Indica la complejidad asintótica en el caso peor del siguiente algoritmo

```
int f(int n, int m){
  int c = 0;
  for (int k = n; k > 51; k -= 1)
    c += 29;
  for (int i = n; i <= n + 19; i += 9)
    for (int j = i - 6; j <= m + 39; j *= 5)
      c -= 28;

 return c;
}
```

Seleccione una:

- a. $\Theta\left(n+\log m\right)$ ✔ Cierto. Hay dos bucles independientes. El primero da siempre un número proporcional a n de vueltas, por lo que su coste es $\Theta\left(n\right)$. Mientras que el segundo es un bucle doble, en el que el bucle de la i da siempre un número constante de vueltas independientemente del valor de n, mientras que el bucle interno, el de la j, da un número de vueltas proporcional a $\log m$ vueltas. Por ello este segundo bucle tiene un coste $\Theta\left(\log m\right)$. Como no sabemos qué valor es mayor, n o m, el coste del código es $\Theta\left(n+\log m\right)$.
- b. $\Theta\left(\log m\right)$
- c. $\Theta\left(n*m\right)$
- d. Ninguna de las anteriores.

a. Cierto. Hay dos bucles independientes. El primero da siempre un número proporcional a n de vueltas, por lo que su coste es $\Theta\left(n\right)$. Mientras que el segundo es un bucle doble, en el que el bucle de la i da siempre un número constante de vueltas independientemente del valor de n, mientras que el bucle interno, el de la j, da un número de vueltas proporcional a $\log m$ vueltas. Por ello este segundo bucle tiene un coste $\Theta\left(\log m\right)$. Como no sabemos qué valor es mayor, n o m, el coste del código es $\Theta\left(n+\log m\right)$.
b. Falso. Hay dos bucles independientes. El primero da siempre un número proporcional a n de vueltas, por lo que su coste es $\Theta\left(n\right)$. Mientras que el segundo es un bucle doble, en el que el bucle de la i da siempre un número constante de vueltas independientemente del valor de n, mientras que el bucle interno, el de la j, da un número de vueltas proporcional a $\log m$ vueltas. Por ello este segundo bucle tiene un coste $\Theta\left(\log m\right)$. Como no sabemos qué valor es mayor, n o m, el coste del código es $\Theta\left(n+\log m\right)$.
c. Falso. Hay dos bucles independientes. El primero da siempre un número proporcional a n de vueltas, por lo que su coste es $\Theta\left(n\right)$. Mientras que el segundo es un bucle doble, en el que el bucle de la i da siempre un número constante de vueltas independientemente del valor de n, mientras que el bucle interno, el de la j, da un número de vueltas proporcional a $\log m$ vueltas. Por ello este segundo bucle tiene un coste $\Theta\left(\log m\right)$. Al ser independientes el coste no es el producto sino la suma $( n + m ).
d. Falso. La respuesta correcta es $\Theta\left(n+\log m\right)$ .

La respuesta correcta es: $\Theta\left(n+\log m\right)$

**Pregunta 8**

Correcta

Se puntúa 1,00 sobre 1,00

En la versión entera del problema de la mochila resuelto con vuelta atrás una cota optimista o beneficio estimado puede ser

Seleccione una:

- a. el valor de la solución parcial más el mínimo de los valores de los objetos multiplicado por el número de objetos que quedan por considerar.
- b. el valor de la solución parcial más el máximo de los valores de los objetos multiplicado por el número de objetos que quedan por considerar. ✔ Cierto. En un problema de maximización del beneficio la cota optimista ha de ser una cota superior de la mejor solución alcanzable. El valor de la solución parcial más el máximo de los valores de los objetos multiplicado por el número de objetos que quedan por considerar es cota superior de la mejor solución alcanzable.
- c. el valor de la solución parcial más el valor del objeto más valioso de los que quedan por considerar.
- d. Ninguna de las anteriores.

a. Falso. En un problema de maximización del beneficio la cota optimista ha de ser una cota superior de la mejor solución alcanzable. El valor de la solución parcial más el mínimo de los valores de los objetos multiplicado por el número de objetos que quedan por considerar no es cota superior de la mejor solución alcanzable.
b. Cierto. En un problema de maximización del beneficio la cota optimista ha de ser una cota superior de la mejor solución alcanzable. El valor de la solución parcial más el máximo de los valores de los objetos multiplicado por el número de objetos que quedan por considerar es cota superior de la mejor solución alcanzable.
c. Falso. En un problema de maximización del beneficio la cota optimista ha de ser una cota superior de la mejor solución alcanzable. Sumar al valor de la solución parcial el valor del objeto más valioso de los restantes no proporciona una cota superior de la mejor solución alcanzable.
d. Falso. La respuesta correcta es: el valor de la solución parcial más el máximo de los valores de los objetos multiplicado por el número de objetos que quedan por considerar.

La respuesta correcta es: el valor de la solución parcial más el máximo de los valores de los objetos multiplicado por el número de objetos que quedan por considerar.

**Pregunta 9**

Incorrecta

Se puntúa -0,33 sobre 1,00

Dados el vector $v = [10, 5, 4, 3, 1]$ y el predicado

$P(v, p, q) = \exists i : p \leq i < q : v[i+1] > v[i]$,

¿Qué devuelve la llamada $P(v, 2, 2)$?

Seleccione una:

- a. El predicado no está bien definido cuando $p = q$. ✖ Falso. El predicado $P$ es cierto si hay un índice $i$ del intervalo $[p, q)$ que cumple que $v[i+1] > v[i]$. Para los valores p = q, el rango $1 \leq i < 1$ es vacío, por lo que el predicado es trivialmente falso independientemente del vector $v$ y el predicado está bien definido.
- b. 1.
- c. 0.
- d. Ninguna de las anteriores.

a. Falso. El predicado $P$ es cierto si hay un índice $i$ del intervalo $[p, q)$ que cumple que $v[i+1] > v[i]$. Para los valores p = q, el rango $1 \leq i < 1$ es vacío, por lo que el predicado es trivialmente falso independientemente del vector $v$ y el predicado está bien definido.
b. Falso. El predicado $P$ devuelve un booleano, no un número entero. El predicado $P$ es cierto si hay un índice $i$ del intervalo $[p, q)$ que cumple que $v[i+1] > v[i]$. Para los valores p = q, el rango $1 \leq i < 1$ es vacío, por lo que el predicado es trivialmente falso independientemente del vector $v$.
c. Falso. El predicado $P$ devuelve un booleano, no un número entero. El predicado $P$ es cierto si hay un índice $i$ del intervalo $[p, q)$ que cumple que $v[i+1] > v[i]$. Para los valores p = q, el rango $1 \leq i < 1$ es vacío, por lo que el predicado es trivialmente falso independientemente del vector $v$.
d. Cierto. La respuesta correcta es: false.

La respuesta correcta es: Ninguna de las anteriores.

**Pregunta 10**

Sin contestar

Se puntúa como 0 sobre 1,00

Queremos calcular el menor de los dígitos de un número $n$ utilizando la siguiente función recursiva generalizada, que utiliza un parámetro acumulador $ac$ para conseguir una versión recursiva final. ¿Cuál puede ser el valor inicial del parámetro $ac$ si utilizamos $n$ como valor del primer parámetro en la llamada inicial?

```
int menorDigito(int n, int ac){
  int resultado;
  if (n<=9) {
    resultado = min(n,ac);
  }else{
    ultDigito = n % 10;
    sigNum = n / 10;
    resultado = menorDigito(sigNum, min(ultDigito,ac));
  }
  return resultado;
}
```

Seleccione una:

- a. 0
- b. 5
- c. 2
- d. Ninguna de las anteriores.

a. False. Si el valor inicial fuese 0, el resultado sería siempre 0 por ser menor o igual que cualquier otro dígito.
b. False. Si el valor inicial fuese 5, el resultado sería incorrecto si todos los dígitos fuesen mayores que 5.
c. False. Si el valor inicial fuese 2, el resultado sería incorrecto si todos los dígitos fuesen mayores que 2.
d. True. La respuesta correcta es: 9

La respuesta correcta es: Ninguna de las anteriores.