Comenzado el	viernes, 10 de enero de 2025, 10:49
Estado	Finalizado
Finalizado en	viernes, 10 de enero de 2025, 10:54
Tiempo empleado	5 minutos 14 segundos
Calificación	**0,00** de 10,00 (**0**%)

Pregunta 1

Sin contestar

Se puntúa como 0 sobre 1,00

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

```
int i = 0, j = n;
while (i < j){
  if (v[i]%10 > v[j]%10)
       j-- ;
  i++;
}
```

Seleccione una:

- a. i.
- b. j.
- c. $n - i - 5$.
- d. Ninguna de las anteriores.

a. Falso. La variable i crece en todas las iteraciones, no puede ser una función de cota.
b. Falso. La variable j no decrece en todas las iteraciones, depende de los valores del vector v. Por ello no puede ser una función de cota.
c. Falso. Si $n \leq 4$, el valor inicial de i es 0, y esta expresión comienza siendo negativa, por lo que no es una cota válida
d. Cierto. La respuesta correcta es: $n - i + 1$.

La respuesta correcta es: Ninguna de las anteriores.

Pregunta 2

Sin contestar

Se puntúa como 0 sobre 1,00

Dado el siguiente código, $0 < r \leq v.size\,(\,)$, $0 \leq f \leq v.size\,(\,)$ y v es un vector de enteros que cumple $\forall i: 0 \leq i < r: v[i] = 0$:

```
int n = 0, i = r, j = 0, a = r;
while (a < f){
   if(v[a]%2 == 0)
     ++i;
   else
     ++j;
   if(v[a-r] == 0)
     --i;
   else
     --j;
   if(i < j)
     ++n;
   ++a;
}
```

¿Cuál de estas propiedades es un invariante de este bucle?

Seleccione una:

- a. $(i = \#k: 0 \leq k < a: v[s]\%2 = 0) \wedge (j = \#k: 0 \leq k < a: v[s]\%2 \neq 0)$
- b. $n = \#p, q: 0 \leq p < q \leq a \wedge q - p = r: P(v, p, q)$, donde $P(v, x, y) = (\#s: x \leq s < y: v[s]\%2 = 0) < (\#s: x \leq s < y: v[s]\%2 \neq 0)$
- c. $r < a < f$
- d. Ninguna de las anteriores.

a. Falso. Incorecto. El bucle calcula el número de intervalos de longitud r tales que el número de valores impares es mayor que el número de valores pares en el vector v y ese valor se guarda en la variable n. Para llevar la cuenta del número de valores pares del intervalo usamos la variable i, mientras que j cuenta el número de variables impares. El intervalo corresponde a los índices $[a - r, a)$ y no $[0, a)$ como indican las expresiones. Por eso no es un invariante del bucle.

b. Cierto. El bucle calcula el número de intervalos de longitud r tales que el número de valores impares es mayor que el número de valores pares en el vector v y ese valor se guarda en la variable n. Por lo tanto, en cada vuelta, en n se guarda el número de intervalos de longitud r desde el inicio del vector hasta el índide del bucle, es decir, la variable a. Además el predicado auxiliar $P(v, x, y)$ indica que en el intervalo $[x, y)$ hay más impares que pares. La respuesta correcta es $n = \#p, q: 0 \leq p < q < a \wedge q - p = r: P(v, p, q)$ con $P(v, x, y) = (\#s: x \leq s < y: v[s]\%2 = 0) < (\#s: x \leq s < y: v[s]\%2 \neq 0)$

c. Falso. Incorrecto. El invariante debe ser cierto antes de comenzar el bucle y al terminar el bucle. Al comienzo $a = r$ y al final $a = f$

d. Falso. La respuesta correcta es: $n = \#p, q: 0 \leq p < q \leq a \wedge q - p = r: P(v, p, q)$, donde $P(v, x, y) = (\#s: x \leq s < y: v[s]\%2 = 0) < (\#s: x \leq s < y: v[s]\%2 \neq 0)$

La respuesta correcta es: $n = \#p, q: 0 \leq p < q \leq a \wedge q - p = r: P(v, p, q)$, donde $P(v, x, y) = (\#s: x \leq s < y: v[s]\%2 = 0) < (\#s: x \leq s < y: v[s]\%2 \neq 0)$

Pregunta 3

Sin contestar

Se puntúa como 0 sobre 1,00

Considerando el algoritmo de ordenación rápida o quicksort para un vector de tamaño mayor que 1, ¿cuál de estas afirmaciones es cierta? `

Seleccione una:

- a. Es un algoritmo de partición.
- b. Su coste en el caso peor es $\Theta(n^2)$.
- c. Su coste en el mejor de los casos es $\Theta(1)$.
- d. Ninguna de las anteriores.

a. Falso. Incorrecta. Quicksort usa un algoritmo de partición para repartir el espacio entre los elementos menores y mayores que el pivote, pero eso es sólo una parte del mismo. Quicksort es un algoritmo de ordenación divide y vencerás.
b. Cierto. El peor caso de quicksort es cuando el pivote no divide el espacio de búsqueda en dos mitades sino que queda en un extremo. Por ejemplo si elegimos de pivote el primer elemento y el vector está ordenado ya. En esta circustancia el coste es $\Theta(n^2)$.
c. Falso. Incorrecta. El mejor caso de quiksort es cuando el pivote divide el espacio de búsqueda en dos mitades. En este caso es $\Theta(n\log n)$, nunca $\Theta(1)$.
d. Falso. La respuesta correcta es: Su coste en el caso peor es $\Theta(n^2)$.

La respuesta correcta es: Su coste en el caso peor es $\Theta(n^2)$.

Pregunta 4

Sin contestar

Se puntúa como 0 sobre 1,00

La tía Josefina tiene un negocio de dulces navideños y necesita repartir P paquetes con sus exquisiteces usando el menor número de camiones de reparto (para ahorrar costes). Todos los camiones de reparto son idénticos: tienen la misma capacidad y su alquiler cuesta lo mismo. Cada paquete tiene un volumen v_i, peso w_i y características diferentes que impiden que podamos meter todos los dulces juntos: por ejemplo, no se puden apilar más de 3 cajas con dulces de hojaldre porque se estropean, etc.

Tu misión es minimizar el dinero que Josefina se gastará usando el menor número de camiones. Como ya sabes mucho de algoritmos te planteas si podrías hacer una poda del espacio de búsqueda. ¿Cuál de las siguientes sería una poda válida?

Seleccione una:

- a. No existe una poda para este tipo de problema
- b. Podar con la cantidad actual de camiones alquilados
- c. Podar considerando la cantidad actual de caminones alquilados + 2
- d. Ninguna de las anteriores.

a. Falso. Para un problema de minimización una poda correcta es el valor de la solución parcial, en este caso la cantidad de camiones alquilados por el momento
b. Cierto. Para un problema de minimización una poda correcta es el valor de la solución parcial, en este caso la cantidad de camiones alquilados por el momento
c. Falso. Eso no funcionaría como poda de minimización ya que estaríamos sobreaproximando la cantidad de caminones que necesitamos, puede ser que no necesitemos 2 camiones más sino sólo 1 o ninguno extra. Una poda de minimización debe ser una cota inferior al coste, no una superior
d. Falso. La respuesta correcta es: Podar con la cantidad actual de camiones alquilados

La respuesta correcta es: Podar con la cantidad actual de camiones alquilados

Pregunta 5

Sin contestar

Se puntúa como 0 sobre 1,00

Indica cuales de las siguiente afirmaciones son ciertas con respecto a los algoritmos divide y vencerás:

Seleccione una o más de una:

- a. Dividen el problema original en subproblemas cuyo tamaño es una fracción del original
- b. Sólo se implementan de forma recursiva
- c. Son más eficientes que los algoritmos no divide y vencerás que resuelven el mismo problema
- d. Dividen el problema original en subproblemas que se pueden resolver en paralelo

a. Cierto. Cada subproblema debe tener como tamaño una fracción del original.
b. Falso. Divide y vencerás es una estrategia de resolución de problemas. Por ejemplo, la búsqueda binaria se puede implementar de forma iterativa o recursiva.
c. Falso. Usar divide y vencerás no garantiza mejorar la complejidad.
d. Cierto. Cada subproblema se debe poder resolver de forma independiente antes de combinar las soluciones para obtener la solución al problema original.

Las respuestas correctas son: Dividen el problema original en subproblemas cuyo tamaño es una fracción del original, Dividen el problema original en subproblemas que se pueden resolver en paralelo

Pregunta 6

Sin contestar

Se puntúa como 0 sobre 1,00

¿Cuál es el coste de la siguiente función recursiva?

```
int f(int n, int k){
    if ( n < 10 )
       return 10*k + 1;
    else if( k < n % 10 )
       return f(n/10, k + 1);
    else
       return f (n/10, k);
}
```

Seleccione una:

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

a. Falso. El parámetro que determina el coste es n. La recurrencia es $T(n) = c_0$ si $n < 10$; $T(n) = T(n/10) + c_1$ si $n \geq 10$, así que por el teorema de la división $T(n) \in \Theta(\log n)$.
b. Cierto. El parámetro que determina el coste es n. La recurrencia es $T(n) = c_0$ si $n < 10$; $T(n) = T(n/10) + c_1$ si $n \geq 10$, así que por el teorema de la división $T(n) \in \Theta(\log n)$.
c. Falso. El parámetro que determina el coste es n. La recurrencia es $T(n) = c_0$ si $n < 10$; $T(n) = T(n/10) + c_1$ si $n \geq 10$, así que por el teorema de la división $T(n) \in \Theta(\log n)$.
d. Falso. La respuesta correcta es: $\Theta(\log n)$

La respuesta correcta es: $\Theta(\log n)$

Pregunta 7

Sin contestar

Se puntúa como 0 sobre 1,00

Dado la siguiente función recursiva:

```
int fun(const vector<int>&v, int i, int f){
    if ( i+1 == f && v[i]%2 != 0)
       return 1;
    else if ( i+1 == f && v[i]%2 == 0)
       return 0;
    else{
      int m = (f+i)/2;
      return fun(v, i, m) + fun(v, m, f);
   }
}
```

Si $v = 11, 0, 2, 5, 1, 9$, ¿Qué devuelve la llamada $fun(v, 0, 5)$?

Seleccione una:

- a. 4
- b. 0
- c. 2
- d. Ninguna de las anteriores.

a. Falso. La función devuelve el número de valores impares que hay en el intervalo [i, f), al llamar con $i = 0$ y $f = 5$. Hay 3 valores impares en ese intervalo: 11, 5 y 1

b. Falso. La función devuelve el número de valores impares que hay en el intervalo [i, f), al llamar con $i = 0$ y $f = 5$. Hay 3 valores impares en ese intervalo: 11, 5 y 1

c. Falso. La función devuelve el número de valores impares que hay en el intervalo [i, f), al llamar con $i = 0$ y $f = 5$. Hay 3 valores impares en ese intervalo: 11, 5 y 1

d. Cierto. La respuesta correcta es: 3

La respuesta correcta es: Ninguna de las anteriores.

Pregunta 8

Sin contestar

Se puntúa como 0 sobre 1,00

Indica cuál de las siguientes afirmaciones es correcta

Seleccione una:

- a. $O(\log n) \supseteq O(n^4)$
- b. $\Omega(n) \subseteq \Omega(n^2)$
- c. $\Theta(n^2) = \Theta(n^3)$
- d. Ninguna de las anteriores.

a. Falso. Por el teorema del límite, como $\lim_{n \to \infty} \frac{n^4}{\log n} = \infty$, por lo que $O(n^4) \nsubseteq O(\log n)$.

b. Falso. Por el teorema del límite, como $\lim_{n \to \infty} \frac{n}{n^2} = 0$, $\Omega(n) \nsubseteq \Omega(n^2)$.

c. Falso. Por el teorema del límite, como $\lim_{n \to \infty} \frac{n^2}{n^3} = 0$, $\Theta(n^2) \nsubseteq \Theta(n^3)$, por lo que no pueden ser iguales.

d. Cierto. La respuesta correcta es: $\Omega(n \log n) \subseteq \Omega(n)$

La respuesta correcta es: Ninguna de las anteriores.

Pregunta 9

Sin contestar

Se puntúa como 0 sobre 1,00

Indica la complejidad del siguiente algoritmo

```
int f(int n, int m){
  int z = 0;
  for (int i = -10; i <= n - 8; i += 2)
    z -= -1;
  for (int j = 1; j < m; j *= 2)
    z -= 7;

 return z;
}
```

Seleccione una:

- a. $\Theta\left(n * \log m\right)$
- b. $\Theta(n * m)$
- c. $\Theta\left(n + \log m\right)$
- d. Ninguna de las anteriores.

a. Falso. Hay dos bucles independientes. El primero da un número proporcional a n de vueltas, por lo que su coste es (n). Mientras que el segundo da un número logarítmico de vueltas con respecto a m, por lo que su coste es (m). Al ser independientes el coste no es el producto sino la suma (n +m), es decir, el máximo entre estas expresiones.

b. Falso. Hay dos bucles independientes. El primero da un número proporcional a n de vueltas, por lo que su coste es (n). Mientras que el segundo da un número logarítmico de vueltas con respecto a m, por lo que su coste es (m). Al ser independientes el coste no es el producto sino la suma (n +m), es decir, el máximo entre estas expresiones.

c. Cierto. Hay dos bucles independientes. El primero da un número proporcional a n de vueltas, por lo que su coste es (n). Mientras que el segundo da un número logarítmico de vueltas con respecto a m, por lo que su coste es (m). Como no sabemos si n o m es mayor, el coste es (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 10

Sin contestar

Se puntúa como 0 sobre 1,00

Tenemos la siguiente función con su especificación:

$P: \{0 \leq k \leq n < v.size\,(\,)\}$

```
int f(const vector<int>& v, const int k, const int n)
```

$Q: \{\#a, b: 0 \leq a < b \leq n \wedge b - a + 1 = k : S\,(\,v, a, b\,)\}$

donde, $S\,(\,v, a, b\,) = \forall j: a \leq j < b: v[j] \leq v[j+1]$.

¿Cuál de las siguientes combinaciones de parámetros de entrada y salida satisfacen esta especificiación?

Seleccione una:

- a. Llamada $f\,(\,[-1, 2, 4, 4, 3, 9, 2], 3, 6\,)$ con resultado 3
- b. Llamada $f\,(\,[1, 1, 9, 0, 3, 4, 4], 0, 3\,)$ con resultado 1
- c. Llamada $f\,(\,[1, 1, 9, 0, 3, 4, 4], 2, 5\,)$ con resultado 4
- d. Ninguna de las anteriores.

a. Falso. Para que la entrada cumpla la precondición los valores k y n deben cumplir que son mayores o iguales que 0 y ser menores que v.size(). En este caso se cumple $0 \leq 3 \leq 6 < 7$. Por otro lado, la postcondición indica que el resultado es el número de intervalos [a, b] tales que $6 = n \geq b > a \geq 0$ de longitud $k = 3$ ($b - a + 1 = 3$). Sólo hay 2 intervalos así: [0, 2] y [1, 3] y no 3. Es resultado no cumple la postcondición.

b. Falso. Para que la entrada cumpla la precondición los valores k y n deben cumplir que son mayores o iguales que 0 y ser menores que v.size(). En este caso se cumple $0 \leq 0 \leq 3 < 7$. Por otro lado, la postcondición indica que el resultado es el número de intervalos [a, b] tales que $3 = n \geq b > a \geq 0$ de longitud $k = 0$ ($b - a + 1 = 0$). No hay valores posibles para a y b que cumplan estas condiciones por lo que la cantidad de intervalos es 0 y no 1. Es resultado no cumple la postcondición.

c. Cierto. Para que la entrada cumpla la precondición los valores k y n deben cumplir que son mayores o iguales que 0 y ser menores que v.size(). En este caso se cumple $0 \leq 2 \leq 5 < 7$. Por otro lado, la postcondición indica que el resultado es el número de intervalos [a, b] con $5 = n \geq b > a \geq 0$ tales que su longitud es k ($b - a + 1 = k$) y en los que los valores del vector son crecientes. Para el vector $[1, 1, 9, 0, 3, 4, 4]$ hay 4 intervalos así. Por ello se cumple también la postcondición

d. Falso. La respuesta correcta es: Llamada $f\,(\,[1, 1, 9, 0, 3, 4, 4], 2, 5\,)$ con resultado 4

La respuesta correcta es: Llamada $f\,(\,[1, 1, 9, 0, 3, 4, 4], 2, 5\,)$ con resultado 4