TEST ELIMINATORIO ENERO 2024

Fecha de Creación	@20 de enero de 2024 13:16
Asignatura	FAL
Fecha de Modificación	@20 de enero de 2024 13:19

Pregunta 1
Correcta
Se puntúa 1,00 sobre 1,00
Marcar pregunta

La función de tiempo $t(n)$ de un algoritmo A satisface la siguiente recurrencia:

$t(0) = 1$

$t(n) = n^2 + n\,log_2(n) + 8t(n/2), n > 0$

Al resolver esta recurrencia aplicando el teorema maestro, tenemos que esta función:

Seleccione una o más de una:

- [x] a. Está en $O(n^3)$. ✔ **Cierto**. La recurrencia se ajusta al patrón de división, con $a = 8$, $k = 2$, y $b = 2$. Por tanto, $a = 8 > b^k = 2^2 = 4$. Como resultado, $T(n) \in O(n^{log_2(8)}) = O(n^3)$.
- [ ] b. Está en $O(n\,log\,n)$.
- [ ] c. Está en $O(n^2 + n\,log\,n)$.
- [ ] d. La recurrencia no puede resolverse mediante el teorema maestro, ya que incluye términos no polinómicos (en particular, $n\,log_2(n)$).
- [ ] e. Está en $O(n^2)$.

a. **Cierto**. La recurrencia se ajusta al patrón de división, con $a = 8$, $k = 2$, y $b = 2$. Por tanto, $a = 8 > b^k = 2^2 = 4$. Como resultado, $T(n) \in O(n^{log_2(8)}) = O(n^3)$.
b. **Falso**. Se obtiene el orden de complejidad $O(n^3)$.
c. **Falso**. Se obtiene el orden de complejidad $O(n^3)$
d. **Falso**. Aunque la parte no recursiva, $n^2 + n\,log_2(n)$, tiene un término con componentes logarítmicos, esta expresión está en $O(n^2)$.
e. **Falso**. Se obtiene el orden de complejidad $O(n^3)$

La respuesta correcta es: Está en $O(n^3)$.

Pregunta 2
Correcta
Se puntúa 1,00 sobre 1,00
Desmarcar

Considérese el siguiente algoritmo:

```
bool algo(int a[], int n) {
    bool resul = true;
    int suma = 0;
    int i = n-1;
    while (i >= 0 && resul) {
       suma += a[i];
       i--;
       resul = (suma >= 0);
    }
    return resul;
}
```

¿Cuáles de las siguientes afirmaciones son ciertas?

Seleccione una o más de una:

- [ ] a. $(\Sigma\, i : 0 \leq i < n : a[i]) - suma$ es una expresión de cota del bucle.
- [ ] b. $suma = \Sigma\, j : 0 \leq j < i : a[j]$ es un invariante del bucle.
- [x] c. i es una expresión de cota del bucle. ✔ **Cierto**. i nunca será menor que -1, y, además, decrece en cada iteración.
- [ ] d. La postcondición puede especificarse como $suma = \Sigma\, i : 0 \leq i < n : a[i]$.
- [x] e. $suma = \Sigma\, j : i < j < n : a[j]$ es un invariante del bucle. ✔ **Cierto**. Al comienzo de cada iteración, en $suma$ estará la suma de todos los elementos en el segmento $a(i..n)$, que es precisamente lo que afirma este predicado. Por supuesto que este invariante es demasiado débil para probar la corrección parcial del algoritmo, pero no deja de ser invariante.

a. **Falso**. Ni siquiera puede garantizarse que esta expresión decrece en cada iteración (porque los valores del vector podrían ser 0, o incluso negativos).
b. **Falso**. El algoritmo acumula la suma desde el final hasta el principio, mientras que este invariante asegura que lo hace desde el principio hasta el final.
c. **Cierto**. i nunca será menor que -1, y, además, decrece en cada iteración.
d. **Falso**. De hecho, no tiene sentido referir una variable local declarada en el cuerpo de la función en la especificación. Esta función comprueba si todas las sumas $a[n-1]$, $a[n-2] + a[n-1]$, $a[n-3] + a[n-2] + a[n-1]$ son no negativas, y este es precisamente el hecho que debería expresar la postcondición.
e. **Cierto**. Al comienzo de cada iteración, en $suma$ estará la suma de todos los elementos en el segmento $a(i..n)$, que es precisamente lo que afirma este predicado. Por supuesto que este invariante es demasiado débil para probar la corrección parcial del algoritmo, pero no deja de ser invariante.

Las respuestas correctas son: i es una expresión de cota del bucle.
, $suma = \Sigma\, j : i < j < n : a[j]$ es un invariante del bucle.

Pregunta 3
Incorrecta
Se puntúa 0,00 sobre 1,00
Marcar pregunta

¿Cuáles de los siguientes problemas pueden resolverse ventajosamente mediante un algoritmo "divide y vencerás"?:

Seleccione una o más de una:

- [x] a. El problema de búsqueda de un elemento en un vector, siempre y cuando dicho vector esté ordenado. ✔ **Cierto**. El algoritmo de búsqueda binaria garantiza un coste logarítmico.
- [ ] b. El problema de ordenación de los elementos de un vector, y además puede garantizarse un coste $O(n\ log\ n)$ en el peor caso.
- [ ] c. Cálculo de la distancia mínima entre cada par de puntos de una nube de puntos.
- [ ] d. El problema de búsqueda de un elemento en un vector.
- [x] e. El problema de ordenación de los elementos de un vector, aunque en el peor de los casos siempre se tendrá un coste cuadrático. ✖ **Falso**. La ordenación por mezcla garantiza un coste $O(n\ log\ n)$ en el peor caso.

a. **Cierto**. El algoritmo de búsqueda binaria garantiza un coste logarítmico.
b. **Cierto**. Mediante el algoritmo de ordenación por mezcla.
c. **Cierto**. Puede obtenerse un algoritmo "divide y vencerás" con coste $O(n\ log\ n)$.
d. **Falso**. Si el vector no está ordenado, no se obtiene ventaja frente a una búsqueda secuencial.
e. **Falso**. La ordenación por mezcla garantiza un coste $O(n\ log\ n)$ en el peor caso.

Las respuestas correctas son: El problema de búsqueda de un elemento en un vector, siempre y cuando dicho vector esté ordenado., El problema de ordenación de los elementos de un vector, y además puede garantizarse un coste $O(n\ log\ n)$ en el peor caso.
, Cálculo de la distancia mínima entre cada par de puntos de una nube de puntos.

Pregunta 4
Correcta
Se puntúa 1,00 sobre 1,00
Marcar pregunta

En relación con los algoritmos de *vuelta atrás* ¿Cuáles de las siguientes afirmaciones son ciertas?

Seleccione una o más de una:

- [ ] a. Los errores cometidos en la aplicación de las podas afectan únicamente a la eficiencia del algoritmo, no a su corrección.
- [x] b. El uso de marcadores es un factor importante de cara a la eficiencia. ✔ **Cierto**. Porque agiliza la comprobación de la viabilidad de soluciones parciales.
- [ ] c. El criterio de viabilidad de soluciones se aplica únicamente a soluciones completas.
- [ ] d. El coste de este tipo de algoritmos siempre es del orden exponencial, o mayor.
- [x] e. La estrategia de generación de soluciones parciales es un factor importante de cara a la eficiencia. ✔ **Cierto**. La estregia de generación puede incidir en el número total de soluciones a explorar, lo que, a su vez, redunda en la eficiencia.

a. **Falso**. Porque una poda mal aplicada puede descartar alternativas que conducen a nuevas y mejores soluciones.
b. **Cierto**. Porque agiliza la comprobación de la viabilidad de soluciones parciales.
c. **Falso**. Si así fuera, la estrategia sería equivalente a la estrategia de "fuerza bruta".
d. **Falso**. Por ejemplo, el algoritmo de "vuelta atrás" que encuentra una salida en un laberinto tiene una complejidad lineal.
e. **Cierto**. La estregia de generación puede incidir en el número total de soluciones a explorar, lo que, a su vez, redunda en la eficiencia.

Las respuestas correctas son: El uso de marcadores es un factor importante de cara a la eficiencia., La estrategia de generación de soluciones parciales es un factor importante de cara a la eficiencia.

Pregunta 5
Correcta
Se puntúa 1,00 sobre 1,00
Marcar pregunta

Supongamos que hemos almacenado un vector de enteros en las n primeras posiciones de un array a. ¿Cuáles de los siguientes predicados especifican que en dicho vector hay únicamente valores impares?

Seleccione una o más de una:

- [ ] a. $\neg\exists i : 0 \le i < n : a[i]\%2 \ne 0$
- [x] b. $(\# i : 0 \le i < n : a[i]\%2 = 0) = 0$ ✔ **Cierto**. El predicado está afirmando que hay 0 pares (por tanto, todos los que hayan serán impares).
- [x] c. $\forall i : 0 \le i < n : a[i]\%2 \ne 0$ ✔ **Cierto**. El predicado está afirmando que todos los valores son impares.
- [ ] d. $(\# i : 0 \le i < n : a[i]\%2 \ne 0) > 0$
- [x] e. $\neg\exists i : 0 \le i < n : a[i]\%2 = 0$ ✔ **Cierto**. El predicado está afirmando que no hay valores pares (por tanto, todos los que hayan serán impares).

a. **Falso**. El predicado está afirmando que no hay valores impares.
b. **Cierto**. El predicado está afirmando que hay 0 pares (por tanto, todos los que hayan serán impares).
c. **Cierto**. El predicado está afirmando que todos los valores son impares.
d. **Falso**. El predicado está afirmando que el número de impares es mayor que 0, pero esto no quiere decir que no haya también pares.
e. **Cierto**. El predicado está afirmando que no hay valores pares (por tanto, todos los que hayan serán impares).

Las respuestas correctas son: $(\# i : 0 \le i < n : a[i]\%2 = 0) = 0$
, $\forall i : 0 \le i < n : a[i]\%2 \ne 0$
, $\neg\exists i : 0 \le i < n : a[i]\%2 = 0$

Pregunta **6**
Correcta
Se puntúa 1,00 sobre 1,00
Desmarcar

Sean $P(x)$ y $Q(x)$ predicados, y $E(X)$ una expresión de tipo numérico. Supongamos, así mismo, que $P(x)$, $Q(x)$ y $E(x)$ están siempre definidas. ¿Cuáles de las siguientes afirmaciones son ciertas?:

Seleccione una o más de una:

- a. La expresión $min\ x : P(x) : E(x)$ siempre está definida.
- b. La expresión $\Sigma\ x : P(x) : E(x)$ siempre está definida. ✔ **Cierto**. El valor de esta expresión se obtendrá sumando $E(x)$ para todas las xs que satisfacen $P(x)$ (si no hay ninguna, su valor será 0).
- c. La expresión $\#\ x : P(x) : Q(x)$ puede estar indefinida.
- d. La expresión $\forall\ x : P(x) : Q(x)$ siempre está definida. ✔ **Cierto**. Será 'false' si hay un x que, satisfaciendo $P(x)$, no satisface $Q(x)$, y 'true' en cualquier otro caso (es decir, cuando todos los x que satisfacen $P(x)$, también satisfacen $Q(x)$, lo que contempla también el caso en el que no haya ningún x que satisfaga $P(x)$).
- e. La expresión $max\ x : P(x) \vee Q(x) : E(x)$ puede estar indefinida. ✔ **Cierto**. Será el caso en el que no hay ningún x que satisfaga $P(x)$ o $Q(x)$.

a. **Falso**. La expresión estará indefinida si no hay ningún x que satisfaga $P(x)$.
b. **Cierto**. El valor de esta expresión se obtendrá sumando $E(x)$ para todas las xs que satisfacen $P(x)$ (si no hay ninguna, su valor será 0).
c. **Falso**. El valor de esta expresión será el número de xs que satisfacen tanto $P(x)$ como $Q(x)$ (si no hay ninguna, su valor será 0).
d. **Cierto**. Será 'false' si hay un x que, satisfaciendo $P(x)$, no satisface $Q(x)$, y 'true' en cualquier otro caso (es decir, cuando todos los x que satisfacen $P(x)$, también satisfacen $Q(x)$, lo que contempla también el caso en el que no haya ningún x que satisfaga $P(x)$).
e. **Cierto**. Será el caso en el que no hay ningún x que satisfaga $P(x)$ o $Q(x)$.

Las respuestas correctas son: La expresión $\Sigma\ x : P(x) : E(x)$ siempre está definida.
, La expresión $\forall\ x : P(x) : Q(x)$ siempre está definida.
, La expresión $max\ x : P(x) \vee Q(x) : E(x)$ puede estar indefinida.

Pregunta **7**
Parcialmente correcta
Se puntúa 0,50 sobre 1,00
Marcar pregunta

Considérese la siguiente función recursiva:

```
int algo(int a, int b) {
    if (b == 0) return 1;
    else if (b%2 == 0) {
       int r = algo(a,b/2); /*1*/
       return r*r;
    }
    else {
       return a * algo(a,b-1); /*2*/
    }
}
```

¿Cuáles de las siguientes afirmaciones son ciertas?

Seleccione una o más de una:

- a. La llamada en `/*2*/` es final.
- b. La función es recursiva simple.
- c. La llamada en `/*1*/` es final.
- d. Hay recursión mutua (indirecta), ya que tiene más de un caso recursivo.
- e. La función es recursiva no final. ✔ **Cierto**. De hecho, las dos llamadas recursivas que tiene son no finales.

a. **Falso**. Tras la llamada queda por hacer la multiplicación.
b. **Cierto**. Cada uno de los dos casos recursivos realiza únicamente una llamada recursiva.
c. **Falso**. Tras la llamada queda por hacer la asignación, y la multiplicación $r * r$.
d. **Falso**. Para que exista recursión mutua debería haber, al menos, dos funciones, que se invoquen la una a la otra, y este no es el caso.
e. **Cierto**. De hecho, las dos llamadas recursivas que tiene son no finales.

Las respuestas correctas son: La función es recursiva simple., La función es recursiva no final.

Pregunta **8**
Correcta
Se puntúa 1,00 sobre 1,00
Desmarcar

Dado un algoritmo iterativo A que tiene como precondición a P y como invariante a I, ¿qué hechos podemos afirmar?:

Seleccione una o más de una:

- a. Que I se cumple cuando A comienza a ejecutarse.
- b. Que I se cumple inmediatamente antes de comenzar las iteraciones. ✔ **Cierto**. Por ser I un invariante del proceso iterativo.
- c. Que I se cumple cuando A termina su ejecución.
- d. Que I se cumple inmediatamente después de terminar las iteraciones. ✔ **Cierto**. Por ser I un invariante del proceso iterativo.
- e. Que P implica a I.

a. **Falso**. Lo normal es que, antes de que comiencen las iteraciones, se haya ejecutando cierto código de inicialización. Es tras la ejecución de dicho código cuando se cumple I.
b. **Cierto**. Por ser I un invariante del proceso iterativo.
c. **Falso**. Puede haber código de finalización, una vez finalizadas las iteraciones.
d. **Cierto**. Por ser I un invariante del proceso iterativo.
e. **Falso**. I no tiene porque cumplirse al comienzo de la ejecución (puede haber código de inicialización).

Las respuestas correctas son: Que I se cumple inmediatamente antes de comenzar las iteraciones.
, Que I se cumple inmediatamente después de terminar las iteraciones.

Pregunta 9
Correcta
Se puntúa 1,00 sobre 1,00
Marcar pregunta

Sean $f(n) = 8\,n\,log_{10}\,n + 2\,n^2$ y $g(n) = 7\,n\,log_{10}\,n + 25\,n$. ¿Cuáles de las siguientes afirmaciones son correctas?

Seleccione una o más de una:

- [x] a. $f(n)$ está en $\Omega(g(n))$. ✔ **Cierto**. $f(n)$ crece más rápido que $g(n)$.
- [ ] b. Tanto $f(n)$ como $g(n)$ están en $O(n\,log\,n)$.
- [ ] c. $f(n)$ está en $\Theta(g(n))$.
- [ ] d. $f(n)$ está en $O(g(n))$.
- [x] e. Tanto $f(n)$ como $g(n)$ están en $O(n^2)$. ✔ **Cierto**. $f(n)$ tiene un orden de crecimiento cuadrático, y $g(n)$ un orden de crecimiento $n\,log\,n$, por lo que también estará en $O(n^2)$.

a. **Cierto**. $f(n)$ crece más rápido que $g(n)$.
b. **Falso**. $f(n)$ tiene un orden de crecimiento cuadrático, mayor que el orden $n\,log\,n$.
c. **Falso**. $f(n)$ crece más rápido que $g(n)$.
d. **Falso**. $f(n)$ crece más rápido que $g(n)$.
e. **Cierto**. $f(n)$ tiene un orden de crecimiento cuadrático, y $g(n)$ un orden de crecimiento $n\,log\,n$, por lo que también estará en $O(n^2)$.

Las respuestas correctas son: $f(n)$ está en $\Omega(g(n))$.
, Tanto $f(n)$ como $g(n)$ están en $O(n^2)$.

Pregunta 10
Correcta
Se puntúa 1,00 sobre 1,00
Marcar pregunta

¿Cuáles de las siguientes afirmaciones son verdaderas?:

Seleccione una o más de una:

- [ ] a. Si $f(n)$ está en $O(g(n))$, entonces $f(n)$ crece igual de rápido que $g(n)$.
- [ ] b. Si $f(n)$ está en $\Omega(g(n))$, entonces $f(n)$ crece, como mucho, tan rápido como $g(n)$.
- [x] c. Si $f(n)$ está en $O(g(n))$, entonces $f(n)$ crece, como mucho, tan rápido como $g(n)$. ✔ **Cierto**. La tasa de crecimiento de $f(n)$ es, como mucho, la de $g(n)$.
- [x] d. Si $f(n)$ está en $\Theta(g(n))$, entonces $f(n)$ está en $O(g(n))$. ✔ **Cierto**. $f(n)$ y $g(n)$ crecerán de la misma forma, por lo que la tasa de crecimiento de $f(n)$ será, como mucho, la de $g(n)$.
- [ ] e. Si $f(n)$ está en $O(g(n))$, entonces $f(n)$ está en $\Theta(g(n))$.

a. **Falso**. La tasa de crecimiento de $f(n)$ es, como mucho, la de $g(n)$, pero puede ser menor.
b. **Falso**. La tasa de crecimiento de $f(n)$ será, como mínimo, la de $g(n)$, pero puede ser mayor.
c. **Cierto**. La tasa de crecimiento de $f(n)$ es, como mucho, la de $g(n)$.
d. **Cierto**. $f(n)$ y $g(n)$ crecerán de la misma forma, por lo que la tasa de crecimiento de $f(n)$ será, como mucho, la de $g(n)$.
e. **Falso**. La tasa de crecimiento de $f(n)$ es, como mucho, la de $g(n)$, pero puede ser menor.

Las respuestas correctas son: Si $f(n)$ está en $O(g(n))$, entonces $f(n)$ crece, como mucho, tan rápido como $g(n)$.
, Si $f(n)$ está en $\Theta(g(n))$, entonces $f(n)$ está en $O(g(n))$.