Universidad del Valle

Primer examen parcial

Fundamentos de análisis y diseño de algoritmos

Carlos Andres Delgado S, Ing *

18 de Octubre 2017

Importante: Se debe escribir el procedimiento realizado en cada punto, con sólo presentar la respuesta, el punto no será válido.

1. Ecuaciones de recurrencia [15 puntos]

Utilizando el método de árboles o iteración, solucione la siguiente ecuación de recurrencia

$$T(n) = 5T(\tfrac{n}{4}) + \tfrac{n}{2}, T(1) = O(n)$$

1) $5T(\frac{n}{4}) + \frac{n}{2}$ $i = 1$

2) $5\left(5T(\frac{n}{4^2}) + \frac{n}{2\times 4}\right) + \frac{n}{2}$

$5^2 T(\frac{n}{4^2}) + \frac{n}{4^1}\times\frac{1}{2} + \frac{n}{2}$ $i = 2$

3) $5^3 T(\frac{n}{4^3}) + \frac{n}{4^2}\times\frac{1}{2} + \frac{n}{4}\times\frac{1}{2} + \frac{n}{2}$

$5^3 T(\frac{n}{4^3}) + \frac{1}{2}\left(\frac{n}{4^2} + \frac{n}{4^1} + \frac{n}{4^0}\right)$ $i = 3$

$\frac{n}{4^i} = 1$ $\quad i = \log_4(n)$

$5^{\log_4(n)} T(1) + \frac{1}{2}\sum_{i=0}^{\log_4(n)-1} \frac{n}{4^i} 2^i$

$n^{\log_4(5)} \times O(n) + \frac{1}{2} n\left(\frac{\left(\frac{1}{4}\right)^{\log_4(n)} - 1}{\frac{1}{4} - 1}\right)$ $\quad \frac{1}{n} \curvearrowright \log_4(\frac{1}{4})$

$\underbrace{n^{1.16} \times n}_{O(n^3)} + \underbrace{\frac{1}{2} n\left(\frac{n^{-3/4} - 1}{-3/4}\right)}_{O(n)}$

Solución $\boxed{O(n^3)}$

2. Divide y vencerás, y Estructuras de datos [30 puntos]

Plantee una solución utilizando divide y vencerás, para contar el número **mínimo** de movimientos necesarios para resolver el problema de las torres de Hanoi.

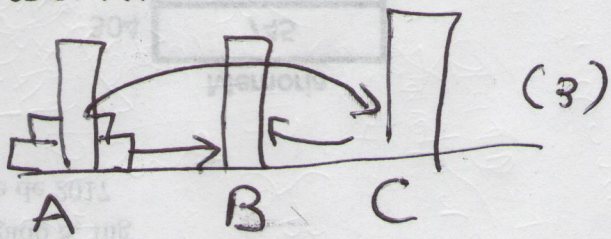
Problema: Este juego consta de n discos y tres postes A,B y C. Inicialmente, el poste A tiene discos de diferente diámetro, acomodados en orden creciente (arriba está el más pequeño y abajo el más grande). La idea es mover los discos al poste B, con la regla de que en los movimientos nunca puede haber un disco más grande que el otro.

Plantee:

- Estrategia de dividir, vencer (solución trivial), y combinar
- Indique una secuencia de pasos o muestre en un pseudocódigo o dibuje un diagrama de flujo para solucionar este problema en un computador. En otras palabras, piense cómo implementaría este algoritmo. **Importante:** Analice que estructuras de datos le ayudarían a solucionar este problema.
- Calcule la complejidad computacional de la solución. Considere las estructuras de datos usadas.

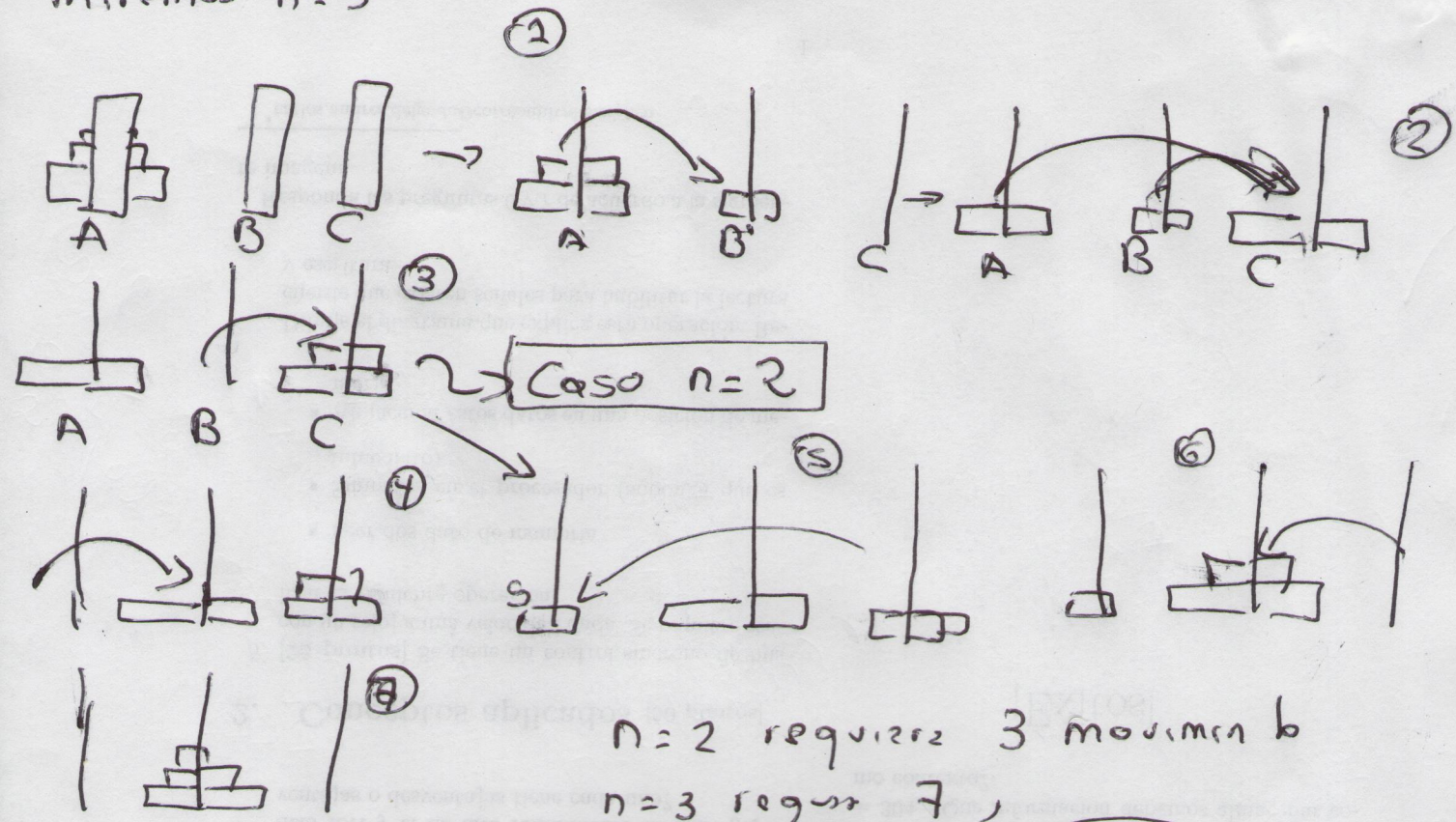
Pista: Para $n = 1$ se necesita 1 movimiento, para $n = 2$ se necesitan 3 movimientos y para $n = 3$ se necesitan 7 movimientos.

Observemos



*carlos.andres.delgado@correounivalle.edu.co

miremos n = 3



n = 2 requiere 3 movimientos

n = 3 requiere 7,

Vemos que $T(n) = 2T(n-1) + 1$

Caso anterior.

$T(1) = 1$

Divide

$T(n)$

Componer

$2 * T(n-1) + 1$

$2(2T(n-2) + 1) + 1$

⋮

Caso trivial → $T(1)$

Conquista

Pides # movimientos, size

```
numHanoi(n)
    if n == 1          } caso trivial
        return 1
    else
        return 1 + 2 * numHanoi(n-1)
    end
```

3. Complejidad computacional [25 puntos]

Indique la complejidad computacional para las siguientes expresiones. Indique para cada linea su numero de ejecuciones en términos de n.

```
1 //Para n > 0
2 for(int i=0; i<=n; i++){
3     for(int j=0; j<=i; j=j+2){
4         //...
5     }
6 }
```

n+2

$\frac{n(n+1)}{4}$

2) $n-0+1+1 = n+2$

$O(n^2)$

3) $T_j \left\lfloor \frac{i}{2} \right\rfloor + 1$

i 0 1 2 3 4 5 6 7

1+1 2+2 3+3 4+4

$\sum_{i=0}^{n} \left\lfloor \frac{i}{2} \right\rfloor + 1 \Rightarrow 1 + \frac{1}{2}\frac{n(n+1)}{2}$

```
1 //Para n > 0
2 for(int i=1; i<=n*n; i=2i){
3     for(int j=i; j<=n; j=j++){
4         //...
5     }
6 }
```

$2\log_2(n)+1$

$2n\log_2(n) - 2n - 1$

$O(n\log(n))$

~~n^2~~

1 2 4 8 16 ... n^2

$\log_2(n)$

0 1 2 3 4 ... $2\log(n)$

$\log_2(n^2)+1$

$\boxed{2\log_2(n)+1}$

i: 1 2 4 8 16 ... n ... x

j: n n-2 n-4 n-8 n-16 ... n-n

$n-2^k \quad 0 \le K \le \log_2(n)$

$\sum_{k=0}^{\log_2(n)} n - 2^k \rightarrow \log_2(n)\,n - \frac{2^{\log_2(n)+1} - 1}{2-1}$

$n\log_2(n) - 2n - 1$

4. Computación iterativa [30 puntos]

Para el siguiente algoritmo:

```
//Para n > 0
algoritmo(n)
    i = 0
    s = 0

    while(i<=2n)
        j = 0
        r = 4

        while(j<=8)
            r += 2
            j++
        end

        s += 2r
        i++
    end
end
```

(j, r) → (j+1, r+2)

(j, 2j+4) → 2(j+1)+4

2j + 4 + 2

r

1. **(15 puntos)** Invariante de ciclo para el ciclo interno y su demostración.

(j, r) - (0, 4) Inicio

(9, 22) Fino

Invariante

$\left(j, \left(\sum_{k=1}^{j} 2\right) + 4\right)$

(j, 2j+4) → 2j+4

2. **(15 puntos)** Invariante de ciclo para el ciclo externo y su demostración

i = 0 s = 0

(i, s) → (0, 0)

(2n+1, 44(2n+1))

(0,0) → (1, 44) → (2, 88)

$\left(i, \sum_{k=1}^{i} 44\right)$

(i, 44i) Invariante.

(0, 0) → (0+1, 0+44)

44(i+1)

44i + 1

Estado anterior.