3/20/20

Monday, March 2, 2020 2:36 PM

HW5) improve bubble

in BubbleSort, after the Kth pass of procedure, k largest elements of the array are located in the last K element of the array and they are in a sorted order (ascending order)

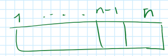
just pseudo code

due MARCH 5

```
for j <- to n-1 do
    if A(j) > A(j+1) then
        swap (A(j), A(j+1))
```

$O(n^2)$

pseudocode



upper limit of loop

```
procedure Binary_Search(A(n), x)
    .
    .
    Binary_Search(A(n/2+1...N), x)
```

⇓

```
procedure BINARY_SEARCH(A(n), x, n)
    .
    .
    Binary_Search(A(2), x, n/2)
```

↑ related to size

Recursive equation of runtime of binary search

$$\begin{cases} T(1)=1 \\ T(n)=T(n/2)+1, & n>1 \end{cases}$$

* $\left[\log_2 n = \log n\right]$ * assume

$T(n) = \log n + 1$

$$\begin{cases} T(1)=1 \\ T(n)=T(n/3)+1, & n>1 \end{cases}$$

find soln

Domain trans

$n=3^K$

$T(n)=T(3^K)=S(K)$

$T(n/3) = T\left(\frac{3^K}{3}\right) = T(3^{K-1}) = S(K-1)$

$T(1) = T(3^0) = S(0) = 1$

$$\begin{cases} S(0)=1 \\ S(K)=S(K-1)+1, & K>0 \end{cases}$$

Telescoping

$S(K)-S(K-1) = 1$

$S(K-1)-S(K-2) = 1$

.
.

$S(1)-S(0) = 1$

$S(K) - S(0) = K\cdot 1$

$S(K) = S(0)+K = K+1$

$n=3^K$

$\log_3 n = \log_3(3^K) = K$

$\boxed{K=\log_3 n}$ substitution

$S(K)=T(3^K)=T(n) = \log_3 n+1$

$T(n) = \longrightarrow$

verify: by induction

basis $n=1$

$T(1) = \log_3 1 + 1 = 1$ (with $\log_3 1 = 0$)

hypothesis: assume $T(K) = \log_3 K+1$

inductive step: must show $T(3K) = \log_3(3K)+1$

$n=3^K$

$K=0$, $K=1$, $K=2$

1, $3^1=3$, $3^2=9$

$T(3K) = T\left(\frac{3K}{3}\right)+1$

$= T(K)+1$

$= (\log_3 K+1)+1$

$= (\log_3 K+\log_3 3)+1$

$T(3K) = \log_3(3K)+1$ ⊠

Must state soln in terms of $\log_2$

$\log_a b = \frac{\log_c b}{\log_c a}$

$\rightarrow T(n) = \log_3 n+1$

$\log_3 n = \frac{\log_2 n}{\log_2 3} = C\cdot\log_2 n$

$C = \frac{1}{\log 3} = \frac{1}{\frac{1}{\log_3 2}} = \log_3 2$

$\log 3 = \frac{\log_3 3}{\log_3 2} = \frac{1}{\log_3 2}$

$\boxed{T(n) = \log_3 2 \lg n+1}$ $\longrightarrow$ verification:

$\Theta(n)$

assume soln $= \log_3 n+1$

$T(n) = \log_3 2 \log n+1$

Basis: $n=1$

$T(1) = \log_3 2 \log(1) + 1 = 1$

inductive hyp assume that $T(K) = \log_3 2 \log_2 K+1$

ind step show that $T(3K) = \log_3 2 \log_2 3K+1$

$T(3K) = T(K)+1$

$= (\log_3 2 \log n+1)+1$

$= (\log_3 2 \log_2 n + \log_2 2)+1$

$= (\log_3 2 \log n + 1\cdot\log 2)+1$

$\left[1 = \log_3 3 = \frac{\log_2 3}{\log_2 3}\right]$

$= \left(\log_3 2 \log n + \frac{\log 3}{\log 3}\right)+1$

$\log 3 = \frac{\log_3 3}{\log_3 2} = \frac{1}{\log_3 2}$

$= \left(\log_3 2 \log n + \frac{\lg 3}{1/\log_3 2}\right)+1$ same

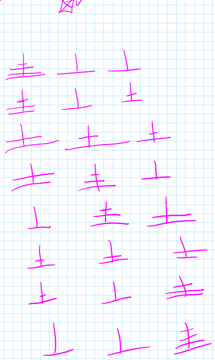
$= \log_3 2(\lg K+\lg 3)+1$

$= \log_3 2 \lg(3K)+1 = T(3K)$ ⊠

Eq which results in which domains involve constants

Towers of hanoi

```
procedure TOH(n, source, aux, dest)
    if n > 0 then
        TOH(n-1, source, dest, aux)
        move 1 disk from source to dest
        TOH(n-1, aux, source, dest)
    end-if
end TOH
```



complexity:

$T(n)$ - the running time of TOH

$$\begin{cases} T(0)=1 \\ T(n) = T(n-1) + \underset{\text{disk movement}}{1} + T(n-1) \end{cases}$$

$$\begin{cases} T(0)=1 \\ T(n) = 2T(n-1)+1, & n>0 \end{cases}$$

$/ \frac{1}{2^n}$ } summation factor

no telescoping bc not same

$\frac{T(n)}{2^n} = R(n)$

$\frac{2T(n-1)}{2^n} = \frac{T(n-1)}{2^{n-1}} = R(n-1)$

$\frac{T(0)}{2^0} = \frac{1}{1} = 1 = R(0)$

Range transf

Range transf

$$\frac{2T(n-1)}{2^n} = \frac{T(n-1)}{2^{n-1}} = R(n-1)$$

$$\frac{T(0)}{2^0} = \frac{1}{1} = 1 = R(0)$$

$$\begin{cases} R(1) = 1 \\ R(n) = R(n-1) + \frac{1}{2^n}, & n > 0 \end{cases}$$

$$\begin{aligned} R(n) - R(n-1) &= \frac{1}{2^n} \\ R(n-1) - R(n-2) &= \frac{1}{2^{n-1}} \\ &\vdots \\ R(1) - R(0) &= \frac{1}{2} \end{aligned}$$

$$R(n) - R(0) = \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^n}$$

$$R(n) = R(0) + \sum_{i=1}^{n} \frac{1}{2^i}$$

$$= 1 + \sum_{i=1}^{n} \frac{1}{2^i}$$

$$R(n) = \sum_{i=0}^{n} \frac{1}{2^i} = \frac{\frac{1}{2^{n+1}} - 1}{\frac{1}{2} - 1}$$

$$= 2 - 2 \cdot \frac{1}{2^{n+1}}$$

$$= 2 - \frac{1}{2^n}$$

$$T(n) = 2^n \cdot R(n)$$

$$= 2^n \cdot \left(2 - \frac{1}{2^n}\right) = 2 \cdot 2^n - \frac{2^n}{2^n}$$

$$\boxed{T(n) = 2^{n+1} - 1}$$

$$a^x a^y = a^{x+y}$$

verification:

Basis $= N = 0$

$T(0) = 2^{0+1} - 1 = 1$ ✓

hyp assume $T(K) = 2^{K+1} - 1$

step : must show $T(K+1) = 2^{K+2} - 1$

$$T(K+1) = 2(T(K)) + 1$$

$$= 2(2^{K+1} - 1) + 1$$

$$= 2 \cdot 2^{K+1} - 2 + 1$$

$$2^{K+2} - 1$$

☒