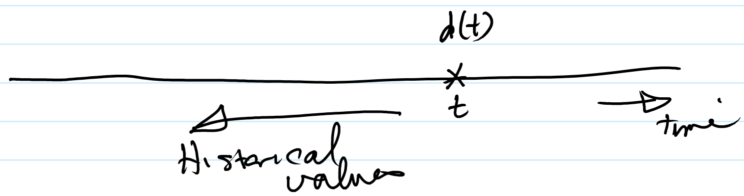
Time Analysis Using Recurrence / difference Equations



difference eqn $\Rightarrow$ $d(t) = 2d(t-1) + 3d(t-2) + t^2$

Recurrence eqn $\Rightarrow$ $d(t) = d(t/2) + 5d(t/5) + e^{-t}$

(1.) Recurrence Eqn Model For Insertion Sort

Model

$T(n)$: Time to Sort an array of size n

$$\boxed{T(n) = T(n-1) + c} \quad \text{--- } (*)$$

c : represent the Effort to insert element #n

Boundary Condition $T(1) = 0$

Sol'n of Eqn $(*)$ means an expression in $\underline{n}$

We have two Special Cases

(1) Last Element is the largest value

$$T(n) = T(n-1) + 1$$
$$T(1) = 0 \quad \text{--- (I)}$$

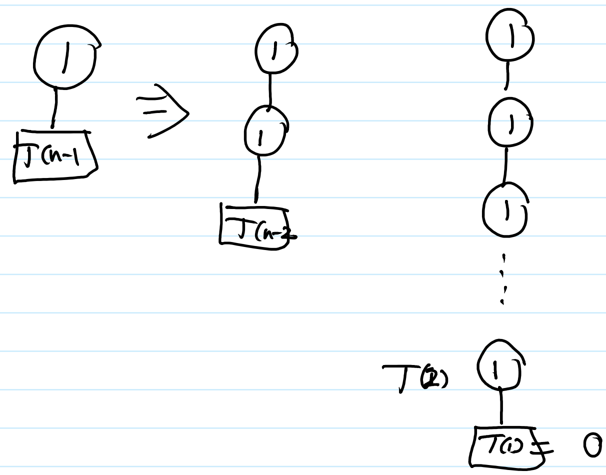
(2) Last Element is the Smallest value . . .

$$T(n) = T(n-1) + n-1$$
$$T(1) = 0$$

<u>Note:</u> Sol'n of difference or Recurrence Equation is done using Recursion Tree

<u>Consider (I)</u>

$T(n) =$



$$T(n) = \underbrace{1 + 1 + 1 + \cdots}_{(n-1)} + 1 + 0$$

$$\boxed{T(n) = (n-1) \quad ; \quad n \geq 1}$$

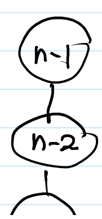
<u>Verification</u>

(1) $T(1) = 1 - 1 = 0$ ✓

(2) $T(n) = T(n-1) + 1$
$= [(n-1) - 1] + 1$
$= n - 1$ ✓

<u>Consider (II)</u> $\quad T(n) = T(n-1) + n-1$
$T(1) = 0$

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



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

$$T(n-2) = n-3$$

$$T(2) = 1$$

$$T(1) = 0$$

$$T(n) = (n-1) + (n-2) + \dots 1$$

$$= (n-1)\left[\frac{n - \cancel{1} + \cancel{1}}{2}\right] = \frac{n(n-1)}{2}$$

$$\boxed{T(n) = \frac{n(n-1)}{2} \;;\quad n \geq 1}$$

Verification

(1) $T(1) = 0$ ✓

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

$$= \frac{(n-1)(n-1-1)}{2} + n-1$$

$$= \frac{(n-1)(n-2)}{2} + \frac{n-1}{2} * 2$$

$$= \frac{(n-1)}{2}\left[n - \cancel{2} + \cancel{2}\right]$$

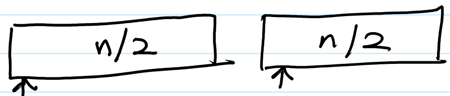
$$= \frac{(n-1)\,n}{2}$$ ✓

Recurrence Eqn Model For DAC Sorting Design

Assume that n which is the size is a power of 2

2, 4, 8, 16, 32, 64, 128

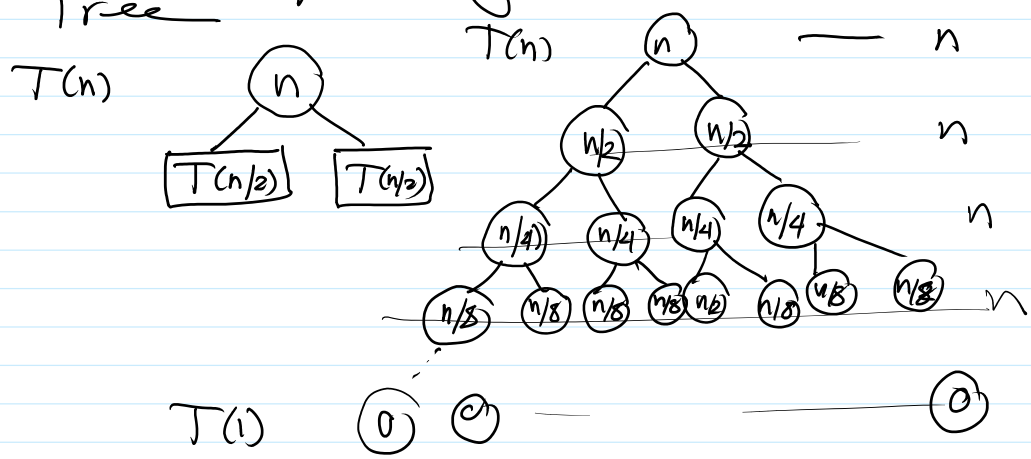
$$\boxed{\quad n \quad}$$



Recurrence Eqn $\Rightarrow$ $\boxed{T(n) = 2T(n/2) + n}$ $\qquad T(2) = 2T(1) + 2$

$T(1) = 0$ Boundary Condition

Let's solve this eqn using Recursion Tree



$$T(n) = \underbrace{n + n + \cdots + n}_{\log_2 n} + 0$$

$$\boxed{T(n) = n\log_2 n}$$

Read 2.1, 2.2, 2.3

HW-1 2.1-1, 2.1-2, 2.1-3, 2.2-1

Due date 01/22/2021