1)

a) $T(n) = 27T(n/3) + n^2$

$T(n) = 27\underbrace{T(n/3)} + n^2$

$T(n/3) = 27T(n/3/3) + (n/3)^2$

$= 27T(n/9) + n^2/9$

$T(n) = 27(27T(n/9) + n^2/9) + n^2$

$729\underbrace{T(n/9)} + 3n^2 + n^2$

$T(n/9) = 27T(n/3/9) + (n/9)^2$

$T(n) = 729(27T(n/27) + n^2/81) + 3n^2 + n^2$

$19683T(n/27) + 9n^2 + 3n^2 + n^2$

General form

$T(n) = 27^k T(n/3^k) + n^2(3^{k-1} + 3^{k-2} + 3^{k-3} \dots 3^0)$

$= 27^k T(n/3^k) + \left[\sum_{i=0}^{k-1} 3^i\right] n^2$

for the $\sum_{i=0}^{n-1} a^i = \frac{1-a^n}{1-a}$

$= 27^k T(n/3^k) + \left(\frac{1-3^k}{1-3}\right) n^2$

$T(1) = 0$

$T(n/3^k = 1) = 0 \quad n = 3^k \quad \log_3 n = k$

$T(n) = 27^{\log_3 n} T(n/3^{\log_3 n}) + \left(\frac{1-3^{\log_3 n}}{2}\right) n^2$

$= \underbrace{n^3 T(n/n)}_{0} + \frac{1}{2}(1-n)n^2$

$= \frac{1}{2} - \frac{n^3}{2} = \Theta(n^3)$

b) $T(n) = 9T(n/4) + n$

$T(n) = 9\underbrace{T(n/4)} + n$

$T(n/4) = (9T(n/16) + n/4$

$T(n) = 81\underbrace{T(n/16)} + (n/4)$

$T(n/16) = 81T(n/16/16) + n/16$

General form

$T(n) = 9^k T(n/4^k) + n\left((\tfrac{9}{4})^{k-1} + (\tfrac{9}{4})^{k-2} \dots \tfrac{9}{4} + \tfrac{9^0}{4^0}\right)$

$9^k T(n/4^k) + n\left(\frac{1-(9/4)^k}{1-9/4}\right)$

$T(n/4^k = 1) = 0 \quad \log_4 n = k$

$T(n) = \underbrace{n^{\log_4 9} T(n/n)}_{0} + (1 - n^{\log_4 9 - \log_4 4}) n \left(-\frac{4}{5}\right)$

$= -5/4 \cdot (n - n^{\log_4 9}) = \Theta(n^{\log_4 9})$

c) $T(n) = 2T(n/4) + \sqrt{n}$

If we apply master theorem (∴)

$2 > 0$ ✓ $4 > 1$ ✓ $1/2 \geq 0$

$\log_b a \Rightarrow \log_4 2 = 1/2 \quad d = 1/2$

$d = \log_b a$

$\Theta(\sqrt{n}\log n)$

d) $T(n) = 2T(\sqrt{n}) + 1$

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

$T(\sqrt{n}) = 2T(n^{1/4}) + 1$

$T(n) = 2(2T(n^{1/4}) + 1) + 1$

$T(n) = 4T(n^{1/4}) + 2 + 1$

$T(n^{1/4}) = 2T(n^{1/8}) + 1$

$4(2T(n^{1/8}) + 1) + 2 + 1$

$= 8T(n^{1/8}) + 4 + 2 + 1$

$= 2^k T(n^{1/2^k}) + (2^{k-1} + 2^{k-2} \dots 2 + 1)$

$= 2^k T(n^{1/2^k}) + \frac{2^k - 1}{2-1}$

I assume $T(2) = 0$

$n^{(1/2)^k} = 2 \quad \log_2 n^{1/2^k} = \log_2 2 \quad = \frac{1}{2^k}\log_2 n = 1 \quad \log_2 n = 2^k$

$k = \log_2 \log_2 n$

$= \underbrace{2^{\log\log n} T(n^{1/2^{\log_2 \log n}})}_{0} + 2^{\log\log n} - 1$

$= (\log n)^{\log_2 2} - 1$

$= \Theta(\log n)$

∴ Master Theorem

$T(n) = aT(n/b) + \Theta(n^d)$

$a > 0,\ b > 1,\ d \geq 0$

$$T(n) \begin{cases} \Theta(n^d) & \text{if } d > \log_b a \\ \Theta(n^d \log n) & \text{if } d = \log_b a \\ \Theta(n^{\log_b a}) & \text{if } d < \log_b a \end{cases}$$

e) $T(n)=2T(n-2)$, $T(0)=1$, $T(1)=1$

$T(0)=1$
$T(1)=1$

$T(2)=2\underset{1}{T(0)} \Rightarrow 2$
$T(3)=2T(1) \Rightarrow 2$
$T(4)=2T(2) \Rightarrow 4$
$T(5)=2T(3) \Rightarrow 4$
$T(6)=2T(4) \Rightarrow 8$
$T(7)=2T(5) \Rightarrow 8$

General form $\Rightarrow$

for all odd numbers $\Rightarrow 2^{\left(\frac{n-1}{2}\right)} = \frac{2^{n/2}}{\sqrt{2}}$

for all even numbers $\Rightarrow 2^{(n/2)}$

so we can say that

$\theta(2^{n/2}) \Rightarrow \theta(2^n)$

f) $T(n)=4T(n/2)+n$, $T(1)=1$

by using master Theorem

$a>0$ $b>1$ $d\geq 0$ $\quad \log_b a = \log_2 4 = 2$

$4>0 \checkmark$ $2>1 \checkmark$ $1\geq 0$

$\underset{2}{\log_2 4} > 1$ $(\log_b a > d)$

$\theta(n^{\log_2 4}) = \theta(n^2)$

g) $T(n)=2T(\sqrt[3]{n})+1$, $T(3)=1$

$T(\sqrt[3]{n}) = 2T(n^{1/9})+1$

$T(n)=2(2T(n^{1/9})+1)+1$

$=4T(n^{1/9})+2+1$

$T(n)=4(2T(n^{1/27})+1)+2+1$

$=8T(n^{1/27})+4+2+1$

General form

$T(n)=2^k T(n^{(1/3)^k}) + \underbrace{(2^{k-1}+2^{k-2}+\dots+2^1+1)}_{\frac{2^k-1}{2-1}}$

$2^k T(n^{(1/3)^k} = 3)$

$n^{(1/3)^k} = 3$

$\log_3 n^{\frac{1}{3}^k} = \log_3 3 = \left(\frac{1}{3}\right)^k \log_3 n = 1$

$\log_3 n = 3^k \quad k=\log_3 \log_3 n$

$2^{\log\log n}\underset{1}{T(3)} + 2^{\log\log n}-1$

$2.\ 2^{\log\log n}-1$

$(\log_3 n)^{\log_3 2}$

$\theta(n^{2^{\log_3\log_3 n}}) \Rightarrow \theta((\log_3 n)^{\log_3 2})$

2)

```
function f(n)
    if n <= 1
        print_line("**")
    else
        for i=1 to n
            f(n/2)
        end for
```

It divides $n/2$ subproblems with size $n/2$ and if we count if → base case (constant

$T(n) = n \cdot T(n/2)+1$

$T(n/2) = n/2\, T(n/4)+1$

$T(n) = \frac{n^2}{2} \cdot T(n/4)+\frac{n}{2}$

$T(n/4) = \frac{n}{4} \cdot T(n/8)+1$

$T(n) = \frac{n^3}{8} \cdot T(n/16) + \frac{n}{2} + \frac{n^2}{8}$

$T(n) = \left(\frac{n}{2}\right)^k \cdot T(n/2^k) + \left(\frac{n}{2}\right)^{k-1} + \left(\frac{n}{2}\right)^{k-2} + \dots \frac{n}{2}+1$

| #for n | #lines |
|---|---|
| 0 | 1 |
| $2^1$ 2 | 2 $2^1$ |
| $2^2$ 4 | 8 $2^3$ |
| $2^3$ 8 | 64 $2^6$ |
| $2^4$ 16 | 1024 $2^{10}$ |
| 32 | 32768 $2^{15}$ |
| 64 | 2097152 |

$2^{k+(k-1)+\dots+1}$ = number of lines

$2^{\sum_{i=0}^{k} i}$

$\boxed{2^{\frac{k \cdot k+1}{2}}}$

we assume $T(1)=1$

$T(1) = n=2^k \quad \log_2 n = k$

$+ \frac{\left(\frac{n}{2}\right)^k - 1}{\frac{n}{2}-1}$

$\left(\frac{n}{2}\right)^{\log_2 n} \cdot \underbrace{T(n/2^{\log_2 n})}_{1} + \frac{\left(\frac{n}{2}\right)^{\log_2 n}-1}{\frac{n}{2}-1}$ $\quad n^{\log n}$

$\frac{n^{\log_2 n}}{n^1} = n^{\log_2 n - 1} = \theta(n^{\log n})$

Scanned by CamScanner

3)

$T(n) = 3T(2n/3) + 1$

It subdives 3 problem with size $2n/3$ and +1 for if

$T(n) = 3T(2n/3) + 1$

$T(2n/3) = 3T((2/3)^2 \cdot n) + 1$

$T(n) = 9T((2/3)^2 \cdot n) + 3 + 1$

General form

$T(n) = 3^k T((2/3)^k n) + \underbrace{3^{k-1} + 3^{k-2} + \dots 3^1 + 3^0}_{\frac{3^k - 1}{3-1}}$

if we assume $T(1) = 1$

$(2/3)^k \cdot n = 1 \quad n = \left(\frac{3}{2}\right)^k \quad \log_{(3/2)} n = k \cdot \underbrace{\log_{(3/2)} (3/2)}_{1}$

$k = \log_{(3/2)} n$

$3^{\log_{3/2} n} T\Big(\underbrace{\underbrace{(2/3)^{\log_{3/2} n} \cdot n}_{1}}_{1}\Big) + \frac{3^{\log_{3/2} n} - 1}{2}$

$3^{\log_{(3/2)} n} + n^{\log_{3/2} 3}$

$n^{\log_{3/2} 3} + n^{\log_{3/2} 3}$

$2 \cdot n^{\log_{3/2} 3}$

$\Theta(n^{2,709})$

4) since all quicksort and insertionsorts are using specific algorithms.

I enter some inputs for both algorithm the results are below

| # of inputsize(1000) | # Swap Operations Quick Sort | Insertion Sort |
|---|---|---|
| 1 | 4777 | 251471 |
| 2 | 5427 | 255922 |
| 3 | 7710 | 253611 |
| 4 | 5409 | 247812 |
| 5 | 5872 | 253597 |

Quicksort Average => 5 839

Insertionsort Average => 252 582.6

It is obvious insertion sort does too many swap operations according to quick sort.

## Quicksort

Assume that its pivot element (L[low]) will be placed in any position after Rearrange

$T = T_1 + T_2$

$T_1$: # of operations in rearrange

$T_2$: # of operations in recursive calls

$A(n) = E[T] = E[T_1 + T_2] = E[T_1] + E[T_2]$

$E[T]$: expected value of T, Expectation linearity

$E[T_1]$: high-low+2 comparisons

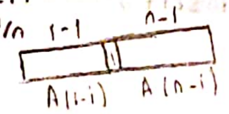
$$E[T_2] = \sum E[T_2 \mid X = x] \cdot \underbrace{P(X = x)}_{= \frac{1}{n} \text{ (equally likely assumption)}}$$

x: Position of the pivot

X: random variable denoting the position of the pivot

RV denoting the position of the pivot

$$A(n) = n+1 + \sum_{i=1}^{n} E(T_2) x = i] . P.(x = i)$$




$$= n+1 + \sum_{i=1}^{n} \underbrace{A(i-1) + A(n-i)]}_{A(0)\ (A+A(1)) + \dots A(n-1) + A(n-2) + A(n-1)} \ ^1/_n$$

$$+ A(n-1)]$$

$$A(n) = n+1 + \frac{2}{n} [A[0] + A(1] + \dots + A(n-1)]$$

$$A(n) = n+1 + 2/n [A[0] + \dots + A(n-2)]$$

$$- \ n.A(n) = n.(n+1) + 2[A[0] + \dots + A(n-2)]$$

$$n A(n) - (n-1) A(n-1) = 2n + 2A[n-1]$$

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

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

change of variable:

$$t(n) = \frac{A(n)}{n+1} \qquad t(n) = t(n-1) + \frac{2}{n+1} \qquad t(1) = t(0) + \frac{2}{2}.1$$

$$t(3) = t(2) + \frac{2}{4} = t(0) + \frac{2}{2} + \frac{2}{3} + \frac{2}{4}$$

$$t(2) = t(1) + \frac{2}{3} = t(0) + \frac{2}{2} + \frac{2}{3}$$

$$t(4) = t(3) + \frac{2}{5} = t(0) + \frac{2}{2} + \frac{2}{3} + \frac{2}{4} + i$$

$\vdots$

$$t(n) = t(0) + 2 \underbrace{\sum_{i=1}^{n} \frac{1}{i+1}}_{O(n)}$$

$$H(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

$$H(n+1) = 1 + \underbrace{\frac{1}{2} + \dots + \frac{1}{n} + \frac{1}{n+1}}_{D(n)}$$

$$H(n+1) - 1$$

$$t(n) = 2 . H(n+1) - 2$$

$$A(n) = t(n).(n+1) = 2(n+1) H(n+1)$$

$$\in \Theta(n \log n)$$

## Quicksort Pseudo

```
Procedure Quicksort(L[low:high])
    if high > low then
        call Rearrange(L[low:high], position)
        call Quicksort(L[low:position-1])
        call Quicksort(L[position+1:high])
    end if
end
Procedure Rearrange(L[low:high] position
    right = low
    left = right + 1
    x = L[low]
    while right < left do
        repeat right = right + 1 until L[right] ≥ x
        repeat left = left - 1 until L[left] ≤ x
        if right < left do then
            call interchange(L[left], L[right])
        end if
    end while
    position = left
    L[low] = L[position]
    L[position] = x
end
```

# Insertion sort

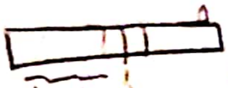
Let $T_i$: # of basic operations at step; $1 \le i \le n-1$

$T = T_1 + T_2 + \dots + T_n$

$A(n) = E[T] = E[T_1 + T_2 \cdots T_n]$ $\qquad E[T] = \sum_{i=1}^{n-1} E[T_i]$

$= E[T_1] + E[T_2] + \dots + E[T_n]$

Calculate $E[T_i] \Rightarrow E[T_i] = \sum_{j=1}^{i} \underbrace{j \cdot P(T_i = j)}_{\text{Prohibit that } \exists j \text{ comparisons in step } i}$



already sorted

There are $i+1$ cases because $\exists i+1$ intervals that $x$ can fall in

$\frac{1}{i+1}$ 1 comparison will occur if $x = L[i] \geq L[i-1]$

$\frac{1}{i+1}$ 2 comparison " " if $L[i-2] < x < L[i-1]$

$\frac{1}{i+1}$ 3 comparison " " if $L[i-3] < x < L[i-2]$

⋮

$\frac{1}{i+1}$ " if $L[1] < x < L[2]$

$\frac{1}{i+1}$ " if $x < L[1]$

$$P(T_i = j) = \begin{cases} \frac{1}{i+1} & \text{if } 1 \le j \le i-1 \\ \frac{2}{i+1} & \text{if } j = i \end{cases}$$

$$E[T_i] = \left( \sum_{j=1}^{i-1} j \cdot \frac{1}{i+1} \right) + \frac{2i}{i+1} = \frac{i(i-1)}{(i+1)2} + \frac{2i}{i+1} = \frac{i^2 - i + 4i}{2(i+1)} = \frac{i(i+3)}{2(i+1)}$$

$$\frac{i(i+3)}{2(i+1)} = \frac{i}{2} + 1 - \frac{1}{i+1} = \frac{i^2 + i + 2i + 2 - 2}{2(i+1)} = \frac{i(i+3)}{2(i+1)}$$

$$A(n) = E[T] = \sum_{i=1}^{n-1} E[T_i]$$

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

harmonix serisi log n gelir n de büyük x bunden $n^2$ de büyük

$$= \frac{n(n-1)}{4} + (n-1) - H(n)$$

$$\in \Theta(n^2)$$

# Insertionsort

```
Procedure InsertionSort(L[1:n])
  for i=2 to n do
    current = L[i]
    position = i-1
    while (position >= 1) and (current < L[position]) do
        L[position+1] = L[position]
        position = position - 1
    end while
    L[position+1] = current
```

★: I used given notes from lecture

right = low

5)

a) $T(n) = 5T(n/3) + n^2$

If we apply master Theorem$(n^d)$

$\log_b a \Rightarrow \log_3 5 \quad d = 2$

$\log_3 5 < 2$

$O(n^2)$

b) $T(n) = 2T(n/2) + n^2$

$\log_b a \Rightarrow \log_2 2 \quad d = 2$

$\log_2 2 < 2$

$O(n^2)$

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

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

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

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

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

$T(n-3) = T(n-4) + n - 3$

$T(n-4) + 4n - 6$

General form

$T(n) = T(n-k) + kn - \sum_{i=0}^{k} i$

We assumed $n-k=0$

$n=k$

$T(n) = T(\underbrace{n-n}_{0}) = n^2 - \left(\frac{n.n+1}{2}\right)$

$\Theta(n^2)$

We can say all relations has same complexity so it does not matter what we will chose if th algorithms have same complexities for other cases.