Divide and Conquer

Concept :-

- Break the large problem into subproblems that are smaller instances of same type of problem
- Recursively solve these subproblems
- Appropriately combine their answers

Examples :-

- Binary search
- Quicksort, Mergesort
- finding the median
- Strassen Matrix Multiplication
- Exponentiation
- Multiplying large integers etc.

⇒ **Finding the minimum no. in set 'S'.**

Given : A set 'S' with 'n' nos.
Goal : find min no. in S'.

Attempt 1 : — sort no. in asc. order. $O(n \log n)$
— Print first no. $O(1)$.

Attempt 2 : — Split set in two equal set S_1 & S_2
— find min. of S_1 & S_2 recursively
— output the minimum of minimum.

$T(n) = 2T(n/2) + 2 \quad T(1) = 0 \quad T(2) = 1$

$O(n)$

Find median in a set of nos

OR

Find k^{th} smallest no. in a set of 'n' nos.

Given :- A set 's' containing 'n' nos

Goal :- Find k^{th} minimum in s.

Ex :- S = { 12, 13, 4, 31, 33, 2, 17, 19, 81, 28, 55, 41, 35, 22, 18, 10, 11, 37, 5, 8, 1, 71, 43, 14, 25 }.

If k = 10, what is output?

Attempt I : Sort no. in asc. order $O(n \log n)$

Print k^{th} no. in list $O(k)$.

Can we do Better than $O(n \log n)$?

Yes, divide and conquer.

1) Split the set 's' into sets of size 5.

12	2	55	10	1
13	17	41	11	71
4	19	35	37	43
31	81	22	5	14
33	28	18	8	25

2) Sort each column

4	2	18	5	1
12	17	32	8	14
(13)	(19)	(35)	(10)	(25)
31	28	41	11	43
33	81	55	37	71

3) Consider the median of the column.

13, 19, 35, 10, 25

4) Find median of these medians.
Call this as Pivot.
10, 13, (19), 25, 35

5) Use Pivot to split set 'S' into S_1 & S_2

$S_1 = \{12, 13, 4, 2, 17, 18, 10, 11, 5, 8, 1, 14\}$ (19).
$S_2 = \{31, 33, 81, 28, 55, 41, 35, 22, 37, 71, 43\}$

6) If $k \leq |S_1|$ then find k^{th} min in S_1
If $k = |S_1|+1$ then output 19.
If $k > |S_1|+1$ then
find $(k-|S_1|-1)^{th}$ min in S_2

Algorithm :- Minimum (S, k)

1. Split S into $\lceil n/5 \rceil$ sets of size 5 each
2. Sort each of these $\lceil n/5 \rceil$ sets.
3. Let $T_1, T_2, \ldots T_k$ be the sets of size 5 with median $m_1 \ldots m_k$.
4. Let 'x' be median of medians.
5. Partition the elements of 'S' into S_1 & S_2 by pivoting at x.
6. If $k \leq |S_1|$ then o/p Minimum (S_1, k)
7. If $k = |S_1|+1$ then o/p 'x'.
8. If $k > |S_1|+1$ then o/p Minimum $(S_2, k-|S_1|-1)$

Analysis

Let $T(n)$ be the Running time for finding the k^{th} minimum in a set of size n.

① ⇒ Split ⇒ $O(n)$

② ⇒ Sort ⇒ $O(n \cdot \frac{n}{5} \cdot 5 \log 5)$

③ ⇒ find median of median ⇒ $T(n/5)$

⑤ ⇒ Split in S_1 & S_2 ⇒ $O(n)$

⑥, ⑧ ⇒ Recursively solve for S_1 or S_2

$T(|S_1|)$ or $T(|S_2|)$.

$T(|S|) = O(n) + T(n/5) + (T(|S_1|) \text{ or } T(|S_2|))$

Can we get upper bound on $|S_1|$ and $|S_2|$

(5)	(4)	(2)	1	18
(8)	(12)	(17)	14	32
(10)	(13)	(19)	25	35
11	31	28	43	41
37	33	81	71	55

y is 1^{st} 2^{nd} 3^{rd} element T_i
$i \leq k/2$ $y < x$

z is 3^{rd} 4^{th} 5^{th} element T_i
$i > k/2$ $z > x$

No. of elements smaller than $x = 3 \cdot k/2 \geq \frac{3n}{10}$.

So $|S_2| \leq 7n/10$. Similarly $|S_1| \leq 7n/10$. — Upper Bound

$T(n) = T(n/5) + T(7n/10) + O(n)$

We can show by induction that $T(n) = O(n)$

⇒ Integer Multiplication -

Given - Two n-bits integers a and b,
Goal - Compute a×b.

Attempt 1 - long Multiplication

```
    1 1 0 1    (13)
  × 1 0 1 1    (11)
  -----------
    1 1 0 1
  1 1 0 1
  0 0 0 0
1 1 0 1
-----------
1 0 0 0 1 1 1 1  (143).
```

$O(n^2)$ complexity.

Attempt 2 Divide and Conquer

To multiply two n-bit integers x and y.
- Divide x and y into low- and high order bits.
- Multiply four $n/2$-bit integers, recursively.
- Add and shift to obtain result.

$$m = \lceil n/2 \rceil$$
$$a = \lfloor x/2^m \rfloor \quad b = x \bmod 2^m$$
$$c = \lfloor y/2^m \rfloor \quad d = y \bmod 2^m$$

$$xy = (2^m a + b)(2^m c + d)$$
$$= 2^{2m} ac + 2^m (bc + ad) + bd.$$

Ex:- $x = \underbrace{1000}_{a}\,\underbrace{1101}_{b}$ $y = \underbrace{1110}_{c}\,\underbrace{0001}_{d}$

Multiply (x, y, n)
If $n = 1$
  Return $x \times y$.
Else.
  $m = \lceil n/2 \rceil$.
  $a = \lfloor x / 2^m \rfloor$ $b = x \bmod 2^m$
  $c = \lfloor y / 2^m \rfloor$ $d = y \bmod 2^m$ } $O(n)$

  $e =$ Multiply (a, c, m)
  $f =$ Multiply (b, d, m)
  $g =$ Multiply (b, c, m)
  $h =$ Multiply (a, d, m) } $4\,T(n/2)$

  Return $2^{2m} e + 2^m (g + h) + f$ } $O(n)$.

$T(n) = 4T(n/2) + O(n)$

Using case 1 of masters theorem - $T(n) = \Theta(n^2)$

Karatsuba trick

To compute middle term $bc + ad$ -

$$bc + ad = ac + bd - (a-b)(c-d)$$

Thus, it requires only 3 $n/2$-bit integers, recursively

$$T(n) = 3T(n/2) + O(n)$$
$$n^{\log_2 3} = \Theta(n^{1.585})$$

Ex :- 2345 0178

a = 23 c = 01

b = 45 d = 78

$(2300 + 45)(0600 + 78)$

$= (23 \times 10^2 + 45)(6 \times 10^2 + 78)$

$= (23 \times 6)\,10^4 + (23 \times 78 + 45 \times 6)\,10^2 + 45 \times 78$

$= ac \times 10^4 + (ad + bc)\,10^2 + bd.$

Similarly, for binary numbers :-

Ex (1) $x = 1101$ $y = 1011$

m = 2 $a = 11$ $c = 10$

$b = 01$ $d = 11$

ac = 11 × 10 = 00 + 11 = 110

bc = 01 × 10 = 00 + 01 = 010

ad = 11 × 11 = 11 + 11 + 11 = 1001

bd = 01 × 11 = 01 + 01 = 011

m = 1	a = 1 b = 1	a = 0 b = 1	a = 1 b = 1	a = 0 b = 1
	c = 1 d = 0	c = 1 d = 0	c = 1 d = 1	c = 1 d = 1
	ac = 1	ac = 0	ac = 1	ac = 0
	bc = 1	bc = 1	bc = 1	bc = 1
	ad = 0	ad = 0	ad = 1	ad = 0
	bd = 0 1	bd = 0	bd = 1	bd = 1

$2^{2m}\,ac + 2^m(bc + ad) + bd$

m = 1			
$= 4 \cdot 1 + 2 \cdot 1 + 0$	$4 \cdot 0 + 2 \cdot 1 + 0$	$4 \cdot 1 + 2 \cdot 2 + 1$	$4 \cdot 0 + 2 \cdot 1 + 1$
$= 6$	$= 2$	$= 9$	3

Then values are returned

$2^{2m}ac + 2^{m}(bc+ad) + bd$

m=2 $= 18.6 + 4(2+9) + 3$

$= 96 + 44 + 3$

$= 143$

Ex (2) $a = 23$ $c = 06$

$b = 45$ $d = 78$

ac	bc	ad	bd
23 x 06	45 x 06	23 x 78	45 x 78 .
a=2 b=3	a=4 b=5	a=2 b=3	a=4 b=5
c=0 d=6	c=0 d=6	c=7 d=8	c=7 d=8
ac = 0	0	14	28
bc = 0	0	21	35
ad = 12	24	16	32
bd = 18	80	24	40 .

m=1 $10^{2m}ac + 10^{m}(bc+ad) + bd$

$100\,ac + 10(bc+ad) + bd$

0+120+18	0+240+30	1400+370+24	2800+670+40
138	270	1794	3510

m=2 $10^{2m}ac + 10^{m}(bc+ad) + bd$

$= 10000 \times 138 + 100(270+1794) + 3510$

$= 1380000 + 206400 + 3510$

$= 1589910$

* Exponential

Ex : a^{27} ⟶ Total 27 multiplication are required.

$= a \cdot a^{26}$

$= a\,(a^{13})^2$

$= a\,(a \cdot a^{12})^2 = a\,(a\,(a^6)^2)^2$

$= a\,(a \cdot ((a^3)^2)^2)^2$

$= a\,(a \cdot ((a \cdot a^2)^2)^2)^2$

Total multiplication required = 7

```
Power (x, y)
{
  if (y == 0)
     return 1
  else if y % 2 == 0 ——(even)
     return Power (x, y/2) * Power (x, y/2)
  else
     return x * Power (x, y/2) * Power (x, y/2)
}
```

Complexity : $T(n) = 2T(n/2) + O(1)$

Case 1 of M.T. ⟹ $\theta(n)$

We can reduce the number of recursive calls made. By using temp. variable

```
Power (x, y)
1. If y==0  return 1
2. else
3       temp = Power (x, y/2)
4       if (y%2 == 0)  ——— even
5           return temp * temp;
6       else
7           return x * temp * temp;
```

Complexity :- $T(n) = T(n/2) + O(1)$

Case 2 of M.T. $\rightarrow$ $\theta(\log n)$