

In C/C++

$\downarrow$   $\downarrow$   $\downarrow$   $\downarrow$   
qsort(arr, n, sizeof(int), cmp);

```
int cmp (const void *a, const void *b)
{
    return *(int*)a - *(int*)b;
}
```

↑  
ascending order

→  $b - a$  ← descending order.

Java

Arrays.sort(arr)

Arrays.sort(arr,

Collections.reverseOrder());

collections.reverseOrder

$$D = \{ \overset{\checkmark}{5}, \overset{\times}{4}, \overset{\times}{3}, \overset{\checkmark}{1} \} \rightarrow \text{not 3x}$$

$$\text{WA} = \underline{\underline{7}}$$

DP

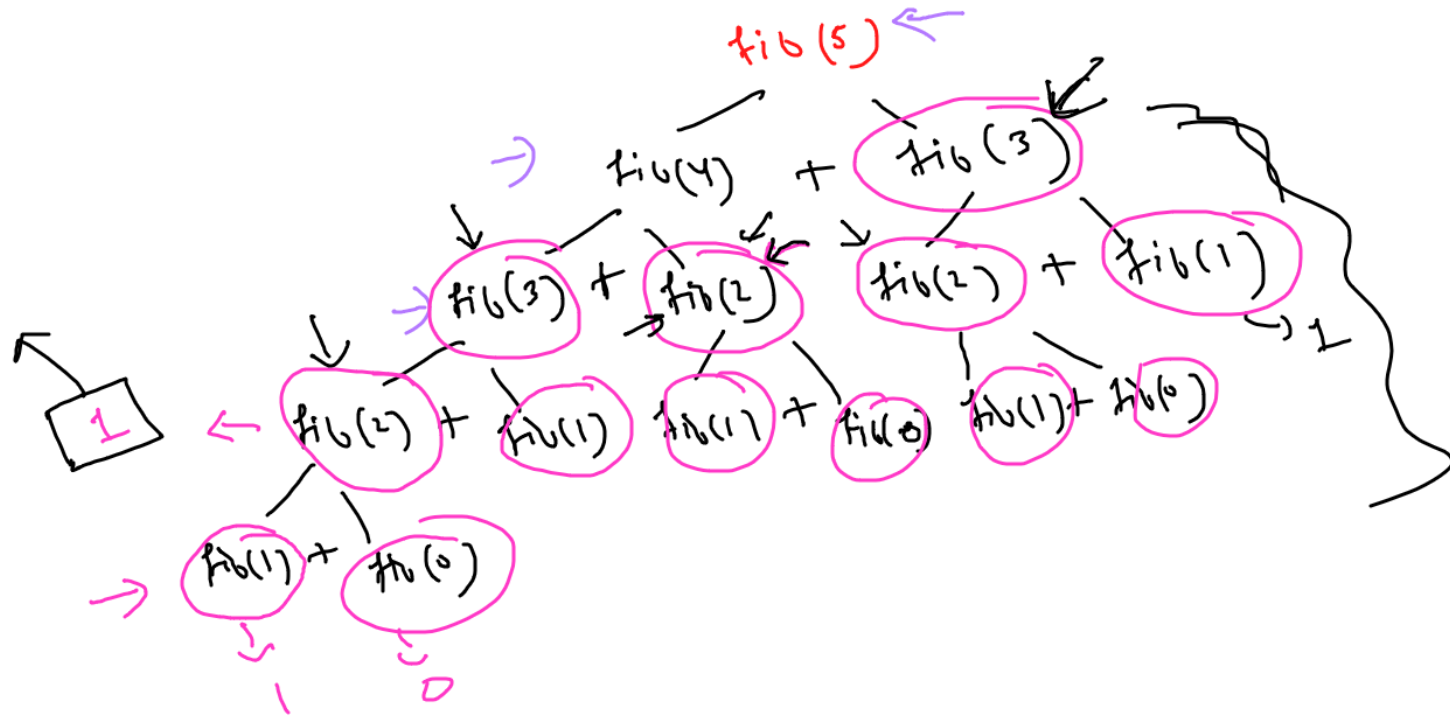
conditions :-

- (1) Problem should be able to be divided into smaller sub-optimal problems.
- (2) Sub-optimal problems should be repeating / overlapping.

principle  $\rightarrow$  Memoization.

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

Divide & Conquer



$$2 \quad D = \{5, 4, 3, 1\}$$

$$wA = 7 \leftarrow$$

6

- $\{1\}$
- $\{1, 3\}$
- $\{1, 3, 4\}$
- $\{1, 3, 4, 5\}$

$$1 + arr[i, R[i] - D[i]]$$

$$7 - 4 = 3$$

$$c = 1 + 1 = 2$$

$$6 - 4 = 2$$

$$c = 1 + 2 = 3$$

$$5 - 4 = 1$$

$$c = 1 + 1 = 2$$

$$6 - 5 = 1$$

$$c = 1 + 1 = 2$$

$$7 - 5 = 2$$

$$c = 1 + 2 = 3$$

		0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8
3	0	1	2	1	2	3	2	3	2
4	0	1	2	1	1	2	2	2	2
5	0	1	2	1	1	1	1	2	2

$$R = \{0, 1, 2, 3, 4, 5, 6, 7\}$$

$$D = \{0, 1, 3, 4, 5\}$$

$$arr[i][j]$$

$$1 + arr[i, 5 - 3]$$

$$1 + arr[i, 2]$$

$$\min(arr[i-1, j], arr[i][j])$$

