* Minimum Number of Coins to reach a particular amount :

Coins = { 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000 }

int V = 91 (Repititions Allowed) result = [ ];

(50, 20, 20, 1)
4 coins

```
for (int i = size-1; i >= 0; i--) {
    while ( V >= coins[i] ) {
        V -= coins[i];
} result.add [ coins[i] ];
```

91 − 2000
91 − 50 = 41
41 − 20 = 21
21 − 20 = 1
1 − 1 = 0 }
res[ ]

31
20
10
1

* Activity Selection Problem :→ (i) Sort (finish) (ii) Select the first always

Given a list of activities with their start & finish times, determine the maximum number of activities that can be completed by a single person at a given time if he/she can perform only one activity at a particular time (without overlapping).

✓(1, 4) (5, 7) (8, 9)

Activity	Start	Finish	Sorted (finish)	O/P
A1	5	7	A3 (1, 4)	A3 (1, 4)
A2	8	9	A6 (3, 5)	
A3	1	4	A5 (0, 6)	A1 (5, 7)
A4	5	9	A1 (5, 7)	
A5	0	6	A2 (8, 9)	
A6	3	5	A4 (5, 9)	A2 (8, 9)

(Lemonade Change Problem) :→

Changes Accepted : $5 $10 $20 (Customers)

Lemonade Price : $5

T1: 5 5 5 10 20

T2: 5 5 10 10 20

20 → 5, 5, 5 }
20 → 10, 5 }

	C	Amount		$5	$10
T1	C1	5	True	1	0
	C2	5		2	0
	C3	5		3	0
	C4	10		2	1
	C5	20		1	0
T2	C1	5	False	1	0
	C2	5		2	0
	C3	10		1	1
	C4	10		0	2
	C5	20			

Sliding Window Problems :→ O/P = 2

Chocolate Distribution Problem :→ (GFG)
(Minimum Absolute Difference)

at least 2

arr = [7, 3, 2, 4, 9, 12, 56]
Packets (chocolates)

m = 3
students

(sort) = [2, 3, 4, 7, 9, 12, 56]
n log n

9 − 4 = 5

7 − 3 = 4

3 m − m
0 — 2 (4) − (2) = (2)

minimum 56 − 4 (Large)

* 2 RoadMaps → (PBC | SBC)

* Bit Manipulation PDF

* Other Resources

* DBMS → SQL

(9:00 − 10:00) PM

(Resources)
(DevOps) VTU LAB Resources

(SDLC) → (Agile) → DevOps → (12 Programs)

(IBM | HCL | TCS | Accenture | Mphasis | Flipkart)

(Ashank)

* (Jan − 2026) → Given an array of only 0's, 1's & 2's sort the array without using any "sorting algorithm".

O(1)

i/p → [2, 1, 1, 0, 2, 1, 2]

o/p → [0, 1, 1, 1, 2, 2, 2]

No extra data structures or built in functions allowed!

for( ) c0 = 1
c1 = 3
c2 = 3 }
for( )

0, 111 222

TC → O(n)

O(n) + O(n) = O(n)