1. Circular Tour
2. Car Fleet
3. Two City Scheduling
4. Candy / Temple offerings
5. Chocolate distribution
6. Queue reconstruction height

Classic Greedy.

1. Biased Standings
2. Defense Kingdom
3. Geo givia
4. Amplifiers
5. Load Balancing

Huffman Encoding & Decoding

1. Circular Tour

Eg:

cost[]:	4	6	7	4
gas[]:	6	5	3	5
idx	0	1	2	3

at 0 → req: 4 get: 6 ✓ Can move to next

at 1 → req: 6 get: 5 + (6−4) = 7 ✓ Can move

at 2 → req: 7 get: 3 + 1 ✗

So, not possible from 0.

at 1 → req: 6 get: 5 ✗

So, not possible from 1. & similarly not possible from 2

at 3 → req: 4 get: 5 → 1 ✓

at 0 → req: 4 get: 7 → 3 ✓

at 1 → req: 6 get: 8 → 2 ✓

at 2 → req: 7 get: 5 → −2 ✗

So, not possible from 3 → Answer does not exist.

Eg:

cost[] :	5	6	7	8	6	4
gas[] :	6	7	4	10	6	5
idx	0	1	2	3	4	5

at 0: $(5,6) \xrightarrow[6-5]{1} (6,7) \xrightarrow[8-6]{2} (7,4) \xrightarrow[6-7]{} -1$ ✗

at 1: $(6,7) \xrightarrow[7-6]{} (7,4) \xrightarrow[5-7]{} -2$ ✗

at 2: $(7,4) \xrightarrow[4-7]{} -3$ ✗

at 3: $(8,10) \xrightarrow[10-8]{2} (6,6) \xrightarrow[8-6]{2} (4,5) \xrightarrow[7-4]{3} (5,6) \xrightarrow[9-5]{4} (6,7) \xrightarrow[11-6]{5} (7,4) \xrightarrow[9-7]{2} (8,10)$

Optimization 1 → Start checking for next interval where the previous tour fails.

Optimization 2 → Take cummulative deficit & check when reaching the last station.
If sum ⪢ prev deficit then circular tour is possible

② Chocolate distribution

Eg: $\underbrace{3 \quad 4 \quad 1}_{\text{Cost} = 4-1=3} \quad 9 \quad \underbrace{16 \quad 7 \quad 9}_{\text{Cost} = 16-7 = 9} \quad 12, \quad M = 3$

✡ Sort. & Pick

win-1, win2, win3

{ 1 3 4 7 7 9 9 12 16 }

Choose all windows of size = 3

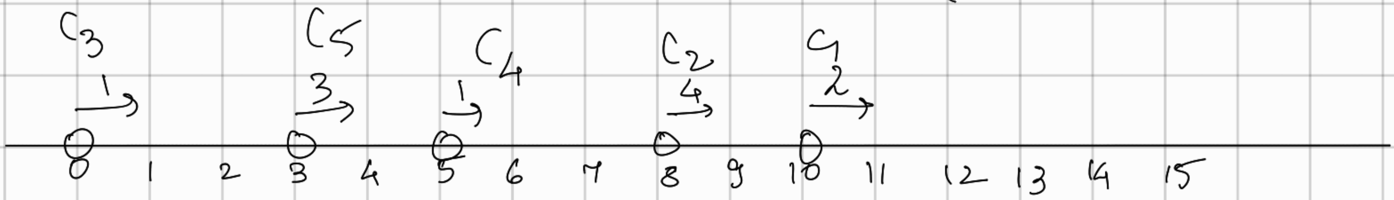
TC: $O(N\log N + (N-M)) = O(N\log N)$

③ Car Fleet.

$[6, 8]$
$\{3, 2\}$

target = 12
positions[] = [10, 8, 0, 5, 3]
speed[] = [2, 4, 1, 1, 3]

$(10-8)/2 = 1s$
$(10-6)/3 = 4/3 = 1.33s$



*: If cars meet at destination, they are also considered as one group.

Observation

1) Right to left Based on distance

$C_1 \Rightarrow$ time $= \frac{12-10}{2} = 1s$
$C_2 \Rightarrow$ time $= \frac{12-8}{4} = 1s$
} 1 group

$C_4 \rightarrow$ time $= \frac{12-5}{1} = 7s$
$C_5 \rightarrow$ time $= \frac{12-3}{3} = 3s$ $(3<7)$
} Same group } 2 group

$C_3 \rightarrow$ time $= \frac{12-0}{1} = 12s$ (>7) } 3 group.

2) Double → TC: $O(N\log N)$

★★

this - other → Min heap or Increasing or Non-decreasing

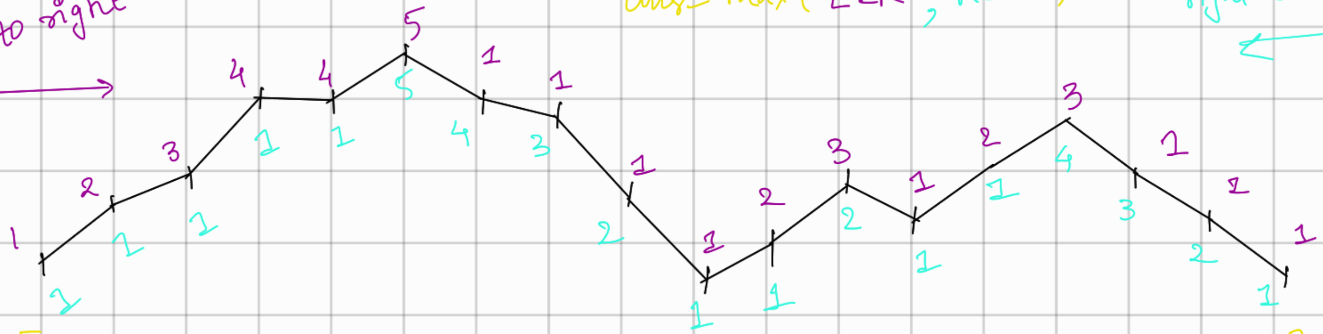
other - this → Max heap or Decreasing or Non-Increasing

④ Candy / Temple Offerings

left to right →

ans = max (L2R , R2L)

right to left ←



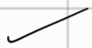
ans[ ] : [1, 2, 3, 4, 4, 5, 4, 3, 2, 1, 2, 3, 1, 2, 4, 3, 2, 1]

Eg: { 10 30 40 70 60 20 }

1 2 3 4 3 2

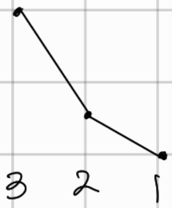
Eg: { 10 40 60 }

ans 1 2 3



Eg: { 50 30 20 }

ans 3 2 1



Eg: { 20 20 20 }

ans 1 1 1

Eg:	{ 10	30	70	60	20 }
ans:	1	2	3	2	1

Eg:	{ 10	30	70	90	100	50	40 }
	1	2	3	4	5	2	1

where order breaks, make min

Eg:	{ 80	90	100	80	60	40	30 }
	1	2	5	4	3	2	1

⑤ Two City Scheduling

{ {10, 20}, {50, 200}, {400, 50}, {15, 20} }

10   150   −350   5

150	{50, 200}
10	{10, 20}
5	{15, 20}
−350	{400, 50}

Sort based on difference b/w {A, B}

- Add first half's B's cost & second half's A's cost to ans.