Problem Solving Paradigms

"If all you have is a hammer, everything looks like a nail"
- Abraham Maslow

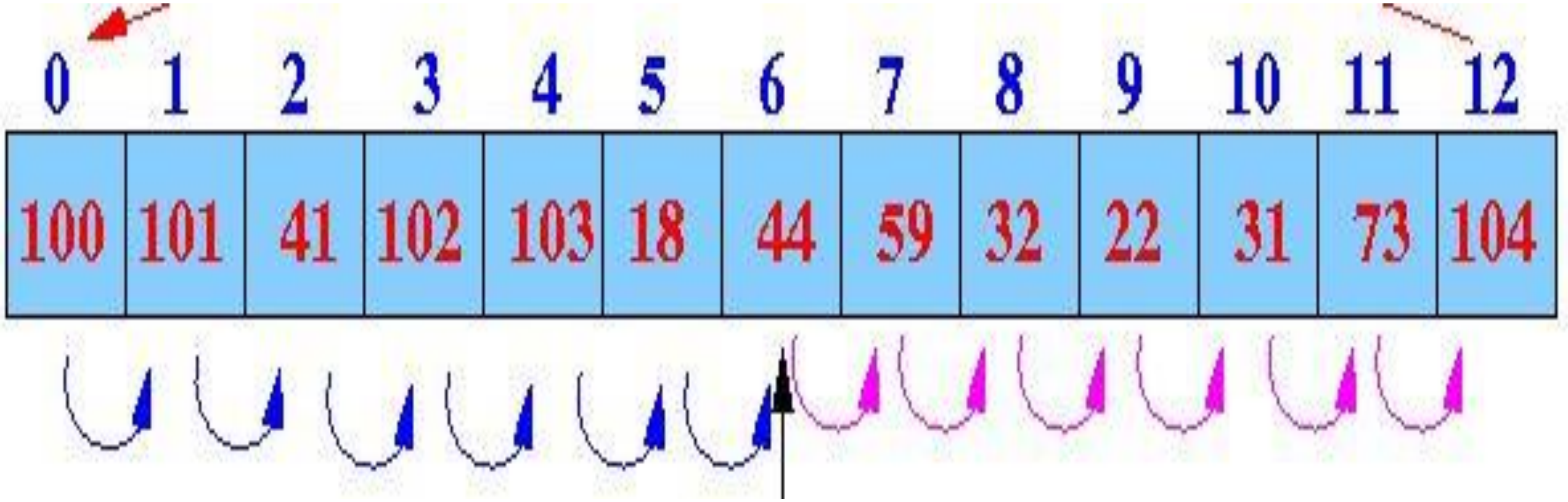
-Esteban Foronda Sierra

Complete Search

- Also know as Brute force
- Should be the first considerated solution
- Should never recive WA(Wrong Answer) response.
- May receive a Time Limit Exceeded(TLE)

What is that?

- Is a method for solving a problem by searching the entire space in bid to obtain the required solution



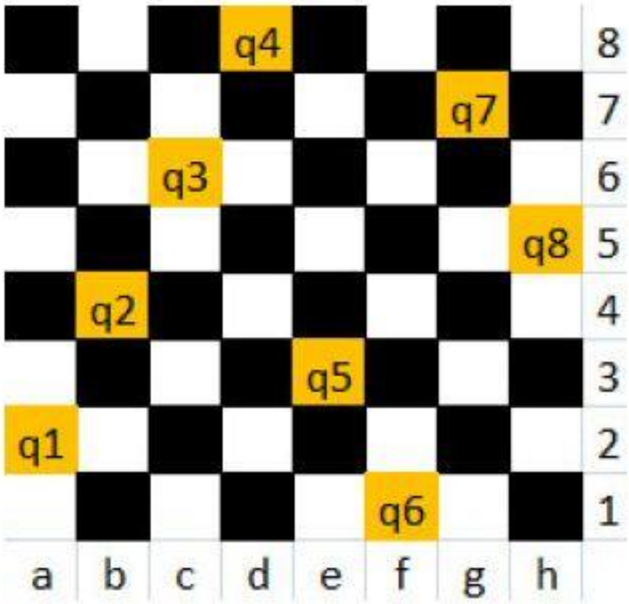
0	1	2	3	4	5	6	7	8	9	10	11	12
100	101	41	102	103	18	44	59	32	22	31	73	104

TLE vs AC

- We also have a few optimization tricks to make some ‘Impossible’ cases become possible.

Queens Chess Problem

- Judge Uva- 750



: One Solution for 8-Queens Problem: {2, 4, 6, 8, 3, 1, 7, 5}

TIPS

- Here are some tips that you may want to consider when designing your solution.

Filtering

- Programs that generate lots of candidate solutions and then choose the ones that are correct(or remove the incorrect ones) are called ‘filters’

Utilize Symmetries

- Some problems have symmetries and we should try to exploit symmetries to reduce execution time.

Prune infeasible Search Space Early

- We may encounter a partial solution that will never lead to a full solution.

Pre-Computation and Pre-Calculation

- Sometime it is helpful to generate tables or other data structures that enable the faster possible lookup of a result. This is called Pre-Computation, in wich one trades memory or space for time.

Try solving the problem Backwards

- Build an attack that looks at the data in some order other than ‘The obvious’

UVa 10360 – Rat Attack

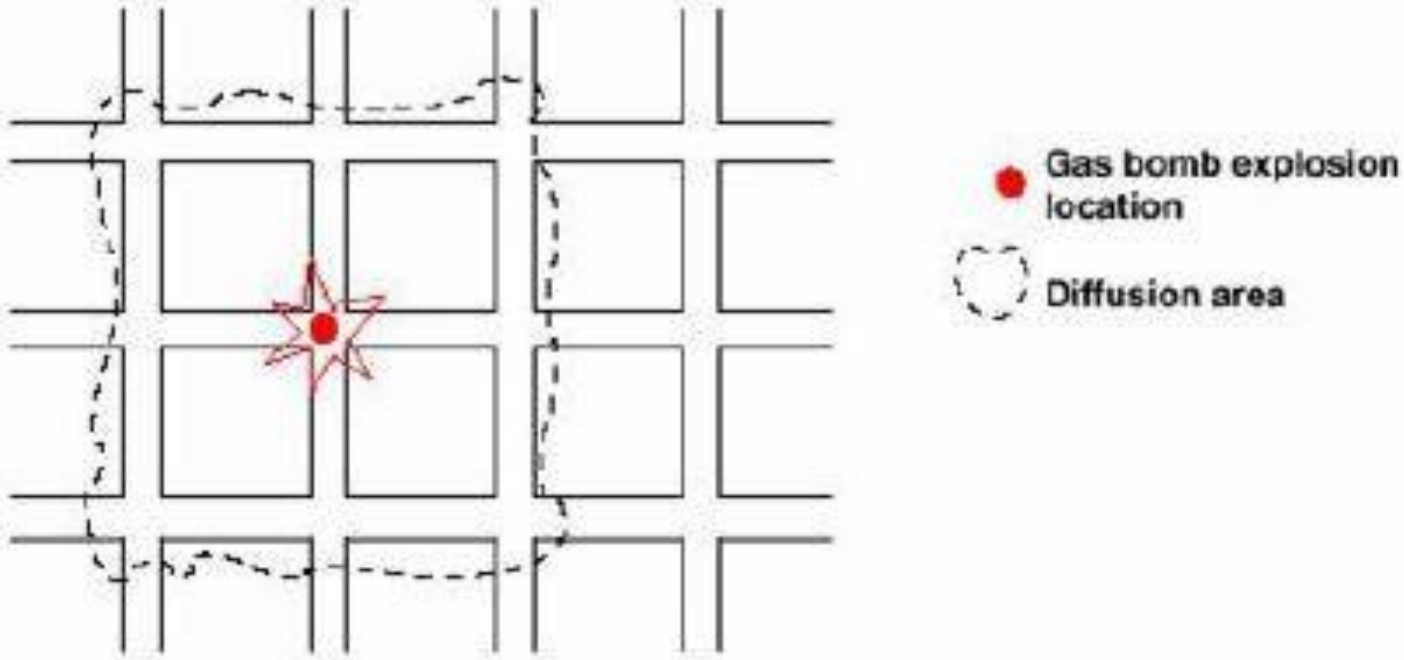
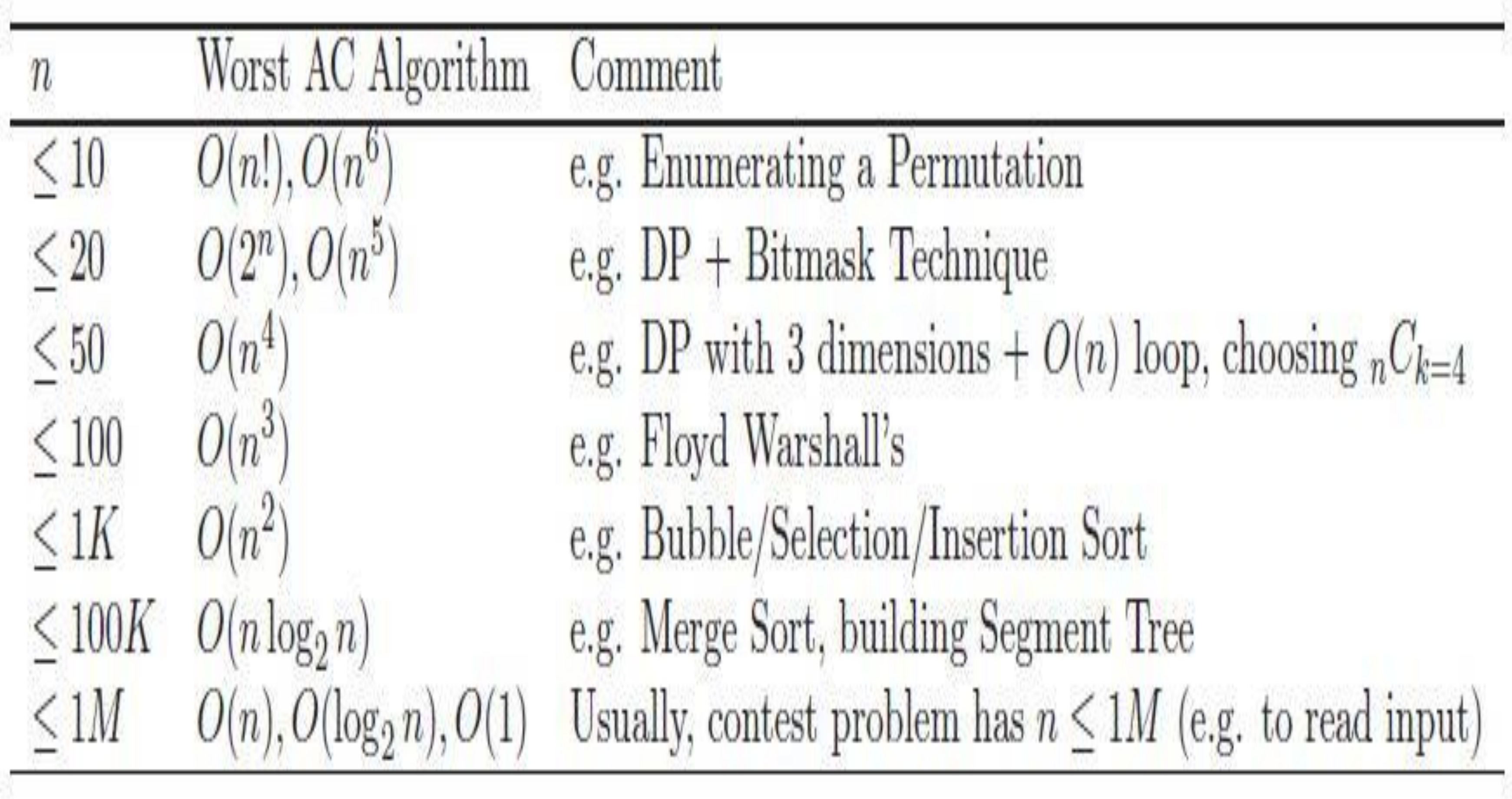


Fig: Rectangular diffusion area of gas bomb

Optimizing Source Code

1) Use scanf/printf rather than cin/cout
2) Use better data structure and algorithm

Max Order

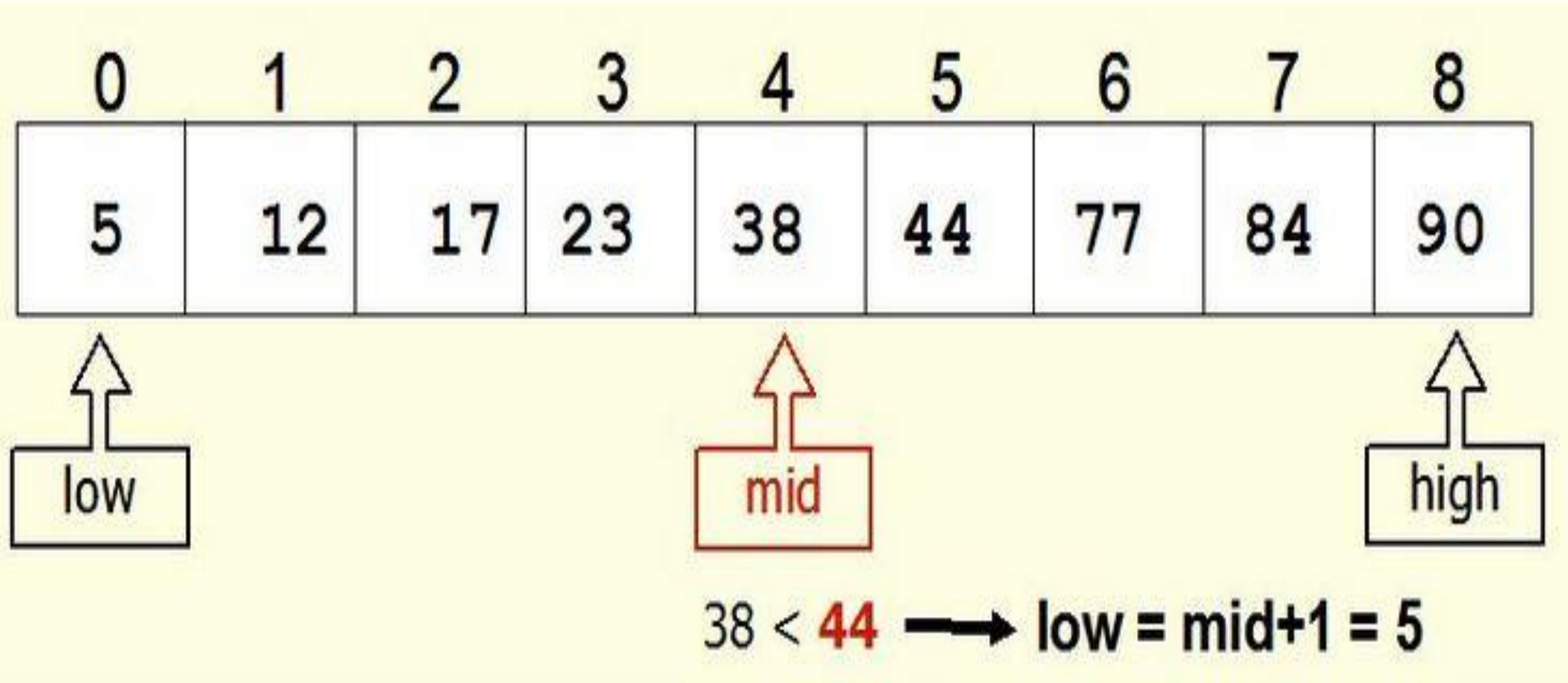


n	Worst AC Algorithm	Comment
≤ 10	$O(n!), O(n^6)$	e.g. Enumerating a Permutation
≤ 20	$O(2^n), O(n^5)$	e.g. DP + Bitmask Technique
≤ 50	$O(n^4)$	e.g. DP with 3 dimensions + $O(n)$ loop, choosing ${}_nC_{k=4}$
≤ 100	$O(n^3)$	e.g. Floyd Warshall's
$\leq 1K$	$O(n^2)$	e.g. Bubble/Selection/Insertion Sort
$\leq 100K$	$O(n \log_2 n)$	e.g. Merge Sort, building Segment Tree
$\leq 1M$	$O(n), O(\log_2 n), O(1)$	Usually, contest problem has $n \leq 1M$ (e.g. to read input)

Divide and Conquer

- Divide the original problem into sub-problems
- Find (sub) Solutions for each of these sub-problems – which are now easier.

Example Binary Search



0	1	2	3	4	5	6	7	8
5	12	17	23	38	44	77	84	90

38 < 44 → low = mid+1 = 5

Greedy

- An algorithm is said to be greedy if it makes locally optimal choice at each step with the hope of finding the optimal solution.

Coins

- {1, 5 , 10, 25} -> 42

- 42 – 25 = 17
- 17 – 10 = 7
- 7 – 5 = 2
- 2 – 1 = 1
- 1 – 1 = 0
- Total: 5 Coins

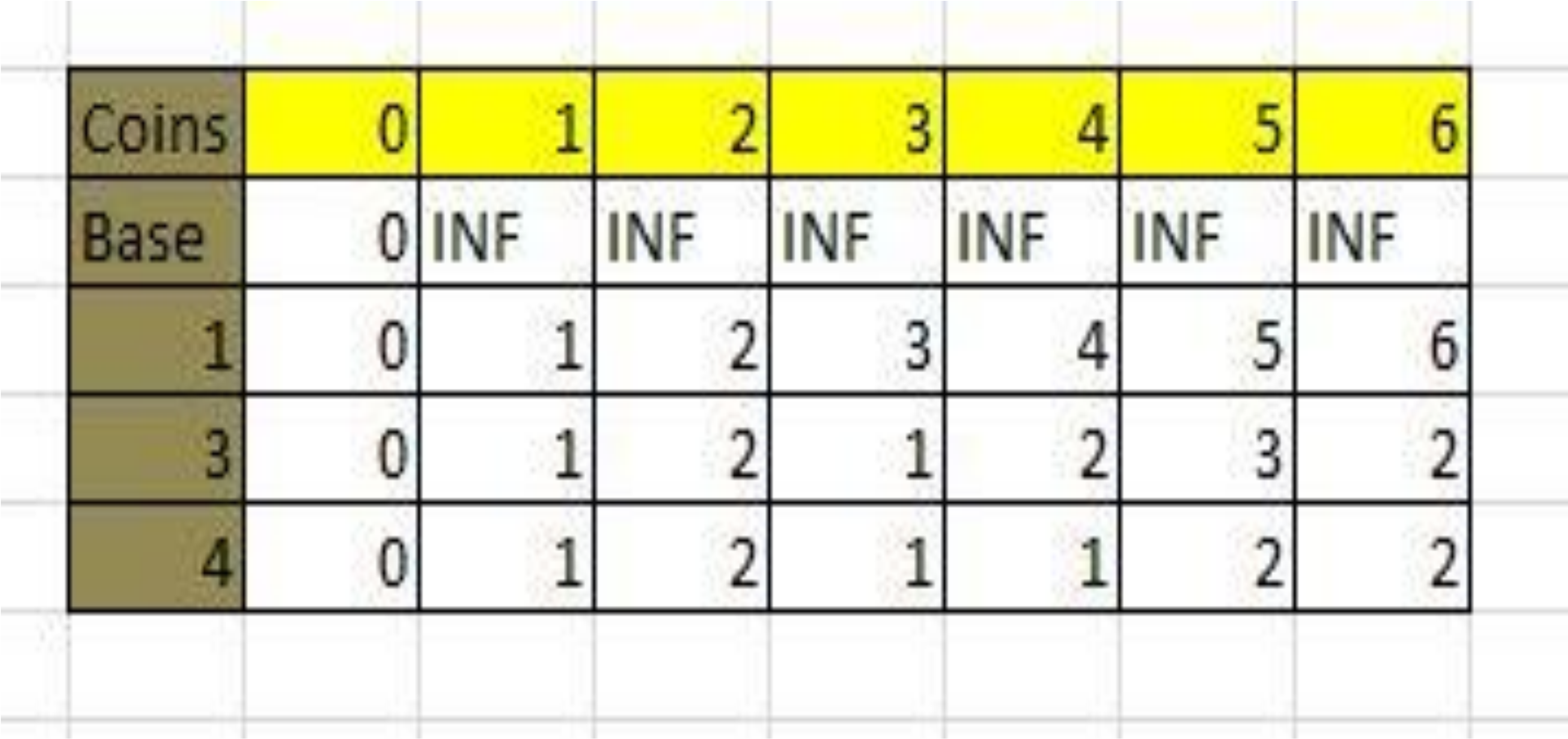
- {1, 3 , 4} -> 6

- ?

Dynamic Programming

- A new result is calculated with a past result.

Coins - DP



Coins	0	1	2	3	4	5	6
Base	0	INF	INF	INF	INF	INF	INF
1	0	1	2	3	4	5	6
3	0	1	2	1	2	3	2
4	0	1	2	1	1	2	2

Bibliography

- Competitive Programming 3 . Steven Halim.
- Competitive Programming 2. Steven Halim.