ICPC Training: Complexity & Binary Search

Coach: Chenyu He

Lesson 01 - Core Mental Models

PREFACE: WHY THIS LESSON?

Coding vs ICPC

- Software Engineering: Feature-first, 'it works' is enough.
- Competitive Programming: Solving under extreme constraints.
-   - Time Limit: Usually 1.0s
-   - Memory Limit: 256MB / 512MB
- Core Conflict:
-   Idea -> Code -> Submit -> TLE (Time Limit Exceeded) -> Panic

The Pro Intuition

- Beginner: Read -> Code immediately -> Too slow -> Try to micro-optimize.
- Pro: Read -> Check Constraints -> Rule out Impossible -> Pick Algorithm -> Code.
- Goal: Map 'N' directly to 'Algorithm' in your brain.

PART 1: COMPLEXITY & ESTIMATION

1 sec = 10^8 ops

- Rule of Thumb: C++ does ~10^8 ops per second.
- Safety check:
- - 10^7: Very Safe
- - 10^8: Risky (tight time limit)
- - 10^9: TLE (almost always)

[Memorize] The N-to-Algorithm Map

N <= 10^5 ~ 2*10^5:

-> O(N log N) (最常见/Most Common)

-> Sort, Binary Search, Heap, SegTree

N <= 5000:

-> O(N^2)

-> Simple DP, All-pairs iteration

N <= 400:

-> O(N^3)

-> Floyd-Warshall, Interval DP

N <= 20:

-> O(2^N * N) (Bitmask DP)

-> O(N!) (Permutations)

N > 10^9 (大整数):

-> O(log N) (Matrix Exponentiation)

-> O(sqrt N) (Number Theory)

Graphs (V, E) ~ 10^5:

-> O(E log V) (Dijkstra)

Memory Pitfalls

- int dp[10000][10000] = 10^8 ints
- -> 10^8 * 4 Bytes = 400MB。
- -> 题目限制 256MB -> MLE (Memory Limit Exceeded)!

- Recursion Depth:
- - Worst case DFS depth = N.
- - At N=10^5, Stack Overflow is likely.

PART 2: BINARY SEARCH

The Essence of Binary Search

- Myth: BS is finding a number in a sorted array.
- Truth: BS is finding the BOUNDARY in a MONOTONIC evaluation space.

- 模型 (Model):
- [False, ..., False, True, ..., True]

- Task: Find the first 'True' or last 'False'.

Coach's Nightmare: Infinite Loops

- BS is short, but dangerous:
- mid = (l + r) / 2;
- if (check(mid)) l = mid; else r = mid;

- 如果 l=4, r=5, mid=4。
- If logic goes l=mid -> l=4 -> Infinite Loop!

- Solution: Stick to ONE template that always shrinks.

[Template] Find First True

-
  ```
  // Find smallest x in [l, r] such that check(x) is true
  long long first_true(long long l, long long r) {
      long long ans = -1;
      while (l <= r) {
          long long mid = l + (r - l) / 2;
          if (check(mid)) {
              ans = mid;
              r = mid - 1; // Try smaller (Left)
          } else {
              l = mid + 1; // Need bigger (Right)
          }
      }
      return ans;
  }
  ```

PART 3: BINARY SEARCH ON ANSWER

When to use BS on Answer?

- 关键词 (Keywords)：
- - '求最小化最大值' (Minimize the Maximum)
- - '求最大化最小值' (Maximize the Minimum)

- 如果你无法直接计算出最优解 (Can't calculate directly)，
- 但是给你一个值 X，你能很快判断 '能不能做到' (Can check feasibility)。
- -> If you can Check(X), you can Binary Search X!

Summary

- 1. Estimate complexity BEFORE coding.
- 2. Memorize one BS template. Understand loop invariants.
- 3. 'Minimize Max' -> BS on Answer.

Selected Homework (Must Do)

- 978C - Letters (Prefix Sum + BS)
- 1476A - K-divisible Sum (Math/BS)
- 1744C - Traffic Light (Observation/BS)
- 474B - Worms (Basic BS)
- 492B - Vanya and Lanterns (BS on Answer)
- 1201C - Maximum Median (Classic BS on Answer! Recommended)