Roadmap

* Get familiar with one of the programming longuge recommended C++ / JAVA (C++ for CP)
  → Do lots of questions on patterns forming to get familiar with loops, make functions, classes etc.
* Start Data Structures & Algos (+ Maths)
  1) Arrays — Most basic but the fundamental data type.
     - Do all famous question on arrays. get familiar with terms such as subarrays, subsequences presum, count array and all.
  2) Do Number Theory (only with help of array ds)
  → It is widely used in most of the questions If you are not familiar with it, then you can't solve that question efficiently.
  3) Get familiar with STL Ds in C++ OR JAVA Liberary (Treeset etc)
  → Use of efficient Ds is must for advanced topics.
  4) Bit Masking / Bit-Manupulation
  → When all above are mastered the must learn Bit operation, base conversions as it is a very scoring topic if you have enough experience & only use ds - arrays.

5) Now start with Complex DS. Tree

→ Get familiar with BST, Rb tree and all other trees and algos such as dfs, bfs, how to find height, depth of a tree.

6) Segment tree / Fenwick Tree

→ Learn both of them to handle range-query question.

Now we have almost covered every ds (except Graphs)

Its time to start Technique practice & Algos.

1) Complete Search
   → Iterative
   → Recursive / Backtracking

2) Divide & Conquer
   → Binary search (Most Important)
   → Fast expo, Matrix expo —

3) Greedy (Already introduced in every topic)
→ This approach is basically based on the observation which part is most useful to us we exploit that and get the answer.

4) Dynamic Programing
→ Every thing from aways to Greedy will help in DP if you are not good in above topics

Then you will suffer a lot in DP.

→ Do 100-200 question of DP to gain experience and little confidence in this topic otherwise you will suffer all the time.

→ It a technique don't try to remember it try to learn how to implement it.

Graphs

→ How to make Graphs
→ Graph traversal algos.
→ Graph type, properties.
→ Minimum Spanning Tree
→ Single Source Shortest Path
→ All - Pair Shortest Path
→ Net work Flow

DSU - Disjoint - Union Set

→ Very useful ds, but only if you know how to use. :)

Now you are done with all the basics in DS & Techniques Time to go deep.

Mathematics
→ Ad - Hoc Maths
→ Combinatorics
→ Probability Theory
→ Cycle Finding
→ Game Theory

String (Advanced)

→ String Matching Algos
  * eg KMP

→ String Matching in 2D Grid

→ String with DP

→ Suffix Trie / Tree / Array

Geometry

→ Basic geometry question based on properties of Polygons of n vertexis.

→ Algos on Polygons
  eg Convex Hull.

If you have completed these topic & you are regular in contests then you are good to go for ~~interviews~~. interviews.

But if wanna do more than that in CP the

→ Backtracking with Bitmask
→ Pruning techniques
→ Meet the Middle Algo
→ DP with Bitmask
⋮

: not to learn but the good part it I will cover that also :). Happy Coding.

Diophantine Eq.

Q = n_1 c_1 n_2 c_2 n

n_1 → type 1 box capacity; c_1 → Its cost; n_2 → Type 2 box capacity; c_2 → Its cost; n → Total object

So we want to find min cost

Let no. of n_1 type boxes be x

Let no. of n_2 type boxes be y

So $n_1 x + n_2 y \leq n$ if (n % gcd != 0) fail

$x', y' \longrightarrow$ by E. Euclid

$$x_0 = \frac{n}{d} \times x'$$

$$y_0 = \frac{n}{d} \times y'$$

$$x = x_0 + \left(\frac{b}{d}\right) p \geq 0$$

$$y = y_0 - \left(\frac{a}{d}\right) p \geq 0$$

$$p \geq -\frac{x_0 d}{b} \quad \text{— Lower limit}$$

$$p \leq \frac{y_0 d}{a} \quad \text{— Upper limit}$$

$$-\frac{x_0 d}{b} \le p \le \frac{y_0 d}{a}$$

Lower — Upper

if (lower > upper)
{ fail }

Now if you iterate over lower to upper to get min cost you will get what TLE :).

Then what to do. → Question will always give you the conditions most of the time to get unique & direct results.

Suppose $c_1/n_1 < c_2/n_2$

So what should be optimal to buy more which cost less i.e c_2/n_2

$$n_1 x + n_2 y = n$$

Type 1 — Type 2 → more to buy this on

$$x = x_0 + \frac{b}{d} p$$

$$\boxed{y = y_0 - \frac{b}{d} p}$$ — To decrease this what should be done

So then for our answer exist at upper limit

Totient function

for a given n count m such that

$gcd(m,n) \neq 1$ & $gcd(m,n) \neq m$

eg $n = 6$

$m = 4$ Ans 1

$n = 2$

$m = 0$ Ans 0