General Inspection

1. Do you Terminate before reading whole input? Do you break while reading, and test case is not fully read?
2. Correct input file / Correct input copy
3. Correctly initialize between test cases
4. Correct reading: E.g. scanf("%d ", &cases); Space is correct in this Input file?
5. Tested corner cases?
6. If you take something from library that does not mean it is 100% right
7. If you take something from library, Revise its comments & prerequisites
8. Validate:
    a. Input stopping conditions
    b. Functions base case / Problem Logic Lines.
    c. No TYPO
    d. Used data types enough to avoid overflow
    e. operators *, ^, /, %
        i. *) /0, %0, /EPS, (n)%x, (-n)%x, +/- EPS
        ii. *) *, ^, counting problems = Overflow
    f. Results fit in 32bit operations, what about intermediate values?
    g. Correct OO Value. Intial value when u maximize & minimize
    h. x%m is X negative? Given Matrix A, find (A^b)%M, Make sure A is initialized correctly in case negatives
    i. if(x%2 == 1) .. What if x negative value?
    j. Set or Multiset?
    k. Wrong pair comparisons: pair1 > pair2, does not check both elements for larger
    l. truncate or approximate, double issues, watch out from -0.0, Floor(-2.3) = -3 but Floor(2.3)= 2
    m. not a number(NAN) which comes from sqrt(-ve), (0/0) ,or cos(1.00000000001) or cos(-1.000000000001)
    n. lp(i, n) lp(j, i+1, n): is there at least 2 elements? Do you need special handling?
9. Precision problems
    a. You should calculate the worst precision. E.g. 1 / 10^9 / 10^5.
    b. Avoid double operations if possible: e.g. integer floor & ceil - even with indepth replacement.
    c. try to do double operations as local as possible
        i. e.g. sum all vs sum part + call(nxt)
    d. Couting the doubles changes little their value
    e. Do binary search to 9 precision, and display x*100. output is only 7 precision
10. BFS with more than one start state
    a. Make sure they are all of same depth
    b. Make sure, In case lexi answer, that you use priority queue
    c. Do we need to validate the intial states?

Wrong Answer

1. Pick a moderated examples, and do problem semantic tracing.
2. Sometimes your added tricky code to make programming easier is just a KILLER BUG
    a. TRY to validate your fancy added code to avoid debugging for silly mistakes
    b. E.g appending a 2d strings array to make it a complete 2d.
        i. Take care, is the appended character part of input? Does it matter?
3. If u have direction array, Does order matters?
4. OVERFLOW
    a. Read numbers N < 2^M where M = 60

    b. Manipulating bit masks with N >= 32, E.g. 1<<40
    c. Multiplications(cross product) & powers & Base conversions.
    d. Is whole code handled for OVERFLOW or it is a mix of int and LL ?!
        i. 1<<x or 1LL<<x
    e. Correct overflow handling
        i. E.g use if(a*b > OO) or if(a > OO/b)
        ii. E.g use if(a+b > OO) or if(a > OO-b)
    f. Input is a 32 integer bit
        i. yes, using int x; will be sufficient, but take care from Integer range
        ii. int x; cin>>x;x = -x; → What if x value = -2^31 --> -x = 2^31 which is OVERFLOW
    g. Exhaustive adding
    h. Final answer fit in 32 bit but intermediate results don't (e.g. polynomial evaluation)
5. Wrong stopping conditions.
    a. Test ends with TestEnd and input with InputFinish. What if such words inside the main input also.
    b. You may need to read 5 numbers if any is valid, u alert
        i. lp(i, 5) { cin>>x; if(!valid(x)) { ok = 0; break;} --> What about output REMINDER?
        ii. Read until number is less than 0, if(n==-1) break --> if(n<0) break;
        iii. Read until x & y & z be zero
        iv. Read until one of x or or z is zero
        v. Read untill Input L, U is L = U = -1
            1. Stop if(L == -1 && U == -1) break;
            2. Check if input like L = -2, U = 3 is valid or not
            3. E.g Number of primes in range [-2,3] = 2
        vi. Each block will be terminated by a line starting with e.
            1. e
            2. egg
        vii. Each block will be terminated by a line containing #.
            1. #
            2. Is this tricky #?
6. Tricky text description
    a. word is "sequence of upper/lower case letters.
        i. This means ali is 1 word, X-Ray is 2 words, ali's book is 3 words
    b. Given 2 integers i, j, find number of primes between them, or in RANGE
        i. Input can be 4 200 OR 200 4
    c. Given N*M grid, Read N lines each start with M chars. E.g. 3*2
        i. 1st line -> ab
        ii. 2nd line -> cdEXTRA // use to depend on read N, M, as RE may happen
        iii. 3rd line -> ef
    d. Do not accept leading zeros numbers?
    e. Do not accept 004, but accept 0 (special case)
7. Geometry
    a. Is there duplicate points? Does it matter? Co-linearity?
8. Graph
    a. Connected or disconnected?
    b. Directed or Undirected?
    c. SelfLoops?
    d. Multiple edges & their effect (MaxFlow sum, SP min)
9. Precision
    a. Watchout -0.0
    b. int x = (int)(a +/- EPS) depends on a > 0 | a < 0.

Time Limit Exceeded

1. May be bug and just infinite loop
2. Can results precomputed in table?!!!!
3. Function calls, may need reference variables.
4. % is used extensively? memset is used extensively?
5. What is blocks of code that represent order? Do we just need to optimize it?
6. Big Input file
    a. Need scanf & printf
    b. Optimize code operations
    c. Switch to arrays and char[]
7. DP Problems
    a. Do you clear each time while it is not needed?
    b. Clear only part of memory u need, not all of memo or use boolean array
    c. The base case order is not O(1)
        i. make sure if(memo != -1) before base case
    d. Use effective base conditions
        i. E.g If you are sure dp(0, M) is X, do not wait untill Dp(0,0)
    e. DP state did not change, so infinite loop
        i. DP(i) call DP(i+s) where s [0-4]
    f. Return result % 10^7, So each time you do operation, you apply %
        i. if DP is huge, change to while(ans >= 10^7) ans -= 10^7
        ii. If mod is 2^p-1, use bitwise
8. BackTracking
    a. If you have different ways to do it, try to do what minimize stack depth
9. Graph problems
    a. Generate dynamic sub-states (edges) only when necessary

Run Time Error

1. Correct input file?
2. Array index out of boundary
    a. Make sure to have correct array size. E.g. If indexing N 1 based, arr[N+1].
    b. Make sure no wrong indexing < 0 || x >= n
        i. Find Primes in range[-2, 3]
        ii. Find factorial -5!
    c. In DP, memo[X][Y], check you access dimensions correctly
    d. In DP, if u have invalid states, make sure to filter them before checking the memo
    e. Stack overflow from infinite recursion
        i. Visited array not marked correctly
        ii. DP with cyclic \ wrong recurrence
    f. You have data structures that requires huge data
    g. /0, %0
    h. Extensive memory allocating until RTE
    i. Using incorrect compare function (e.g. return that return (A, B) same answer as (B, A))
    j. Use uninitialized data: int x; v.resize(x); cin>>x;
    k. Watchout, if multiset contains (3 3 3 3 6 9) and u delete 3 -->will be (6, 9)
    l. To delete one item, use iterator to find & delete it
    m. struct T { int A[]; };