

# Tips

---

## Understanding

---

- Read Q carefully, catch [keywords]
  - [Input] assumptions/conditions/constraints
    - foundation of reasoning
    - or fill incomplete info/gap
  - [Output] requirements
  - [How] methods
  - Compare *definitions* with *purposes* (NOT the same!)
    - clear declaration VS intention or objective

## Analyse

---

- Find a pattern by numerating from **easy examples**.
- Case Split
  - Consider complement as well if simpler.
  - Consider empty, 0, 1, infinity and other *special cases*.
- Use some diagrams, tables, graphs and other colors to help understanding.

## Math

- Don't SKIP necessary steps during calculation
- Simplify
  1. Extract common terms and put unrelated stuff in Const [Const] [Var(x,y,z)Only]
  2. Don't forget Const when integration
- Substitution
- Check Divide by 0, multiple both sides by 0 [CaseSplit!]
- Graph: undefined, key points, asymptotes, limit (0, infinity); domain, codomain, odd and even.

## Express ideas

---

- **Pencil** for code or diagram, leave spare lines for improvement.
- Define what you use.

Let  $x$  to be  $\langle \dots \rangle$

- Ceiling and Flooring when needed  $\lceil x \rceil$ ,  $\lfloor x \rfloor$  [Q with high marks]
- Write more to explain your idea when needed
- Use some diagrams, tables, graphs and other colors for expression.
- Calculate the final results for the ease of marking.
- Have a summary when needed

## Proof

Assume ... (1)

RTP:

By (1) and (2), we know ..., we get ...

Hence, ...

This corresponds to C, as required.

- Making sure that **conditions/assumptions are met**

*Decision on whether disproof or proof*

- Analyse with intuition.
- Disproof
  - Give a solid counter-example satisfying all the conditions, but not the conclusions given.
- Proof
  - Contradiction when conditions are scarce.
  - Induction, state (IH) with  $\phi$ 
    - Mathematical induction on  $N$
    - Rule induction
    - Structural induction (more inversion needed)
    - algorithms, Regex, etc.

## Coding

---

- Have a clear design of algorithm
  - Input and Output
  - Data structure details
  - Complexity
- Assertion to ensure invariant.
- Case Split:
  - Special Case:
    - [Segment Fault] NULL assert(isRight==right);
    - [Infinity Loop] Condition for termination
      - Tree node : NULL/ Leaf

## Debugging

---

- Unit Test for accuracy.
- Make the test case simpler
- printf / step-by-step
- Rethink the problem, check what is being asked.
- Random guessing ...
- Test it before real operation for accuracy. [OS]