Those are the exact lines where you'd plug in your new logic:

- Build a list of **all** subtractive occurrences (IV, IX, XL, XC, CD, CM)
- Sort them by index
- Enforce "next subtractive must have a **lower** value, otherwise invalid"
- Mark all their positions in processedConversion with correct indices

Everything else (rules 2/3/5/6/7, non-adjacent checks, etc.) can stay as-is for now — the core structural fix lives in *this* indexOf cluster.

You said:

Design a small targeted test list of nasty numerals like this to stress your code

ChatGPT said:

You said:

but please also give gentle examples first so that I can implement code. Just give invalid examples since I am certain all my test cases are correct... Hence this is why I stored all the values in array at top of my code

ChatGPT said:

Nice, this is exactly the kind of torture-suite thinking you want for this kind of validator 😄

I'll give you **only invalid examples**, starting gentle and then getting nastier. For each, I'll briefly note *why* it's invalid so you can line it up with your rules and your new subtractive-group logic.

You can literally drop these into a String[] invalidTests = {...} and loop over them.

---

1. Gentle invalids – single, obvious rule breaks

These will help you confirm your basic rule handling still works after you change the subtractive code.

1. IIII
    - 4× I in a row → violates "no more than 3 identical numerals in succession".
2. XXXX
    - 4× X in a row → same rule as above.
3. CCCC

- 4× C in a row → same rule again.

4. VV
    - V cannot repeat; it's a "5" numeral and must appear at most once.
5. LL
    - L cannot repeat.
6. DD
    - D cannot repeat.
7. IL
    - Illegal subtractive: I can only go before V or X.
    - People sometimes misuse this for 49; proper form is **XLIX**.
8. IC
    - Illegal subtractive: I cannot go before C.
    - Often misused for 99; proper is **XCIX**.
9. XD
    - Illegal subtractive: X cannot subtract from D; only from L or C.
    - Misused for 490; proper is **CDXC**.
10. XM
    - Illegal subtractive: X cannot subtract from M.
    - Misused for 990; proper is **CMXC**.
11. VX
    - Illegal subtractive: V may never be used as a subtractive numeral.
    - 15 should be **XV**, not "5 before 10".
12. LC
    - Illegal subtractive: L may not be subtractive; 50 cannot come before 100 to subtract.
    - 50 before 100 must be **L + C (150, CL)**, not a subtractive.

---

2. Medium invalids – small but structurally wrong

These are still short, but will hit combinations of your rules (2, 4, 5, 6, 7).

13. IIV
    - Looks like "1 + (5 – 1)" or "(1 – 1) + 5" if someone is confused.
    - But subtractive patterns must be **exactly one smaller + one larger**; IIV is not allowed.
    - Correct for 3 is III; for 4 is IV.
14. VIV
    - Naively "5 + (5 – 1)" or "(5 – 1) + 5"; also sometimes misused for 9.
    - Correct 9 is **IX**, not VIV.
15. XXL
    - Someone trying to do "20 before 50" subtractively.
    - But X subtracts from L (40) as **XL**, not XXL.
    - 30 is XXX; 70 is LXX. XXL should be rejected.
16. IXX
    - Often misused for 19. Correct 19 is **XIX**.
    - Here the small I is in the wrong position relative to the X's.
17. XLL
    - L can't repeat. Whether you interpret XL as 40 or not, the second L is illegal.
18. DCD
    - Tries to do something like 900+? But proper 900 is CM,
      and D (500) cannot be used as subtractive… and also appears twice here.
19. ILX
    - Mixed illegal subtractive: I before L, then X after.
    - Should fail your "I can only go before V/X" and triangle rules (Rule 7 / non-adjacent checks).

---

3. Nasty invalids – stress your subtractive & indexOf logic

These are the ones especially aimed at:

- multiple subtractive notations
- repeated subtractive notations
- non-decreasing subtractive values
- index-of-only-first-occurrence weaknesses
- overlapping interpretations

You can use these to really torture your indexOf + processedConversion code.

3.1 Repeated subtractive pairs of equal value

20. IVIV
    - Two subtractive IV groups in a row (4 + 4).
    - Proper 8 is VIII.
    - Your validator should reject **repeated subtractive pairs of the same value**.
21. IXIX
    - Two IX pairs (9 + 9 = 18).
    - Proper 18 is XVIII.
    - Good to test "multiple IX" detection and indexOf behaviour.
22. XCXC
    - Two XC (90 + 90 = 180).
    - Proper 180 is CLXXX.
    - Hits exactly the "indexOf only sees first XC" problem.
23. CMCM
    - Two CM (900 + 900 = 1800).
    - Proper 1800 is MDCCC.
    - Very similar structurally to the MCMCMXCIX case you already saw.

3.2 Subtractives in the wrong value order (non-decreasing)

These are designed to test your idea:

"If the next subtractive notation is the same or higher value → error."

24. IXCM

- Subtractive values: IX = 9, CM = 900.
- Values go 9 → 900 (increasing) → should be **invalid**.
- Canonical 909 is CMIX.

25. XCIXCM
    - Subtractive values: XC = 90, IX = 9, CM = 900.
    - Sequence: 90 → 9 → 900 (drops then jumps up) → violates strictly decreasing rule.
26. CMXCXC
    - CM (900), XC (90), XC (90 again).
    - Not only repeated XC, but also second XC not lower than previous subtractive (same value).

3.3 The "classic monster" you already saw

27. MCMCMXCIX
    - Intended as something like 1000 + 900 + 900 + 90 + 9.
    - Canonically, that would be 2899, which should be written MMDCCCXCIX.
    - This is the one that **exposed your current bug**:
        - only first CM seen by indexOf,
        - unTotal hacks,
        - ends up accepted as valid with wrong decimal.

3.4 Overlapping / confusing subtractives

28. XIXIX
    - Can tempt a parser into overlapping IX pairs or mis-grouping ("XI" + "XIX", or "X" + "IX" + "IX").
    - Good for testing that you don't double-count or overlap subtractive pairs.
29. XCXCI
    - Contains XC (90), then XC again overlapping potential region, plus final I.
    - Naive indexOf("XC") once could leave you in a weird processedConversion state.
30. CXCXC

    - CXC can be 190 (100 + 90), but repeated XC or mis-grouping C + XC + XC is structurally dubious.
    - Excellent case to ensure you don't allow multiple equal subtractives of the same "decade".

If you want a **minimal starter subset** to wire your new "subtractive groups must strictly decrease" logic, I'd start with:

- Gentle sanity: IL, IC, XD, XM
- Medium structure: IIV, VIV, XXL, IXX
- Nasty subtractive set: IVIV, IXIX, XCXC, CMCM, IXCM, MCMCMXCIX