arr : [3,1,2]

↳ permutations => 3! = 6.

```
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1
```

① For 312
next permutation is
321.

② 321
↓ next permutation
123

→ Naive Soln:

→ Generate all permutations
→ Perform linear search. [For finding 312]. → O(N! × N)
→ Next index will be the answer. //

→ Optimal Soln :

[2, 1, 5, 4, 3, 0, 0]

→ match every this's

↳ 2 1|5 4 3 00

→ 2154 30 0

→ 21543 0 0

→ 2154 - - - < 003 } both are smaller than 300
                   030

→ 215 - - - - < 3400 - < less
             4030 → less than 4300
             4300 → equal
                ↓

→ 21 - - - - - 5430 } won't give any soln / highest is the
              5340   becoz 5. There won't be
              5 - -   any this ie 75.

① Ray } longer prefix match
  Rax   J & X
  Rbx   → Rax appears
  —   after Ray in
        dictionary

$2 \cdots$

0 — ⎫
1 — ⎬ less than or equal two
3 —

4 — ⎫ we need to choose the
5 — ⎬ minimal element as to set
       ⎭ next permutation

→ 3 ↓ min
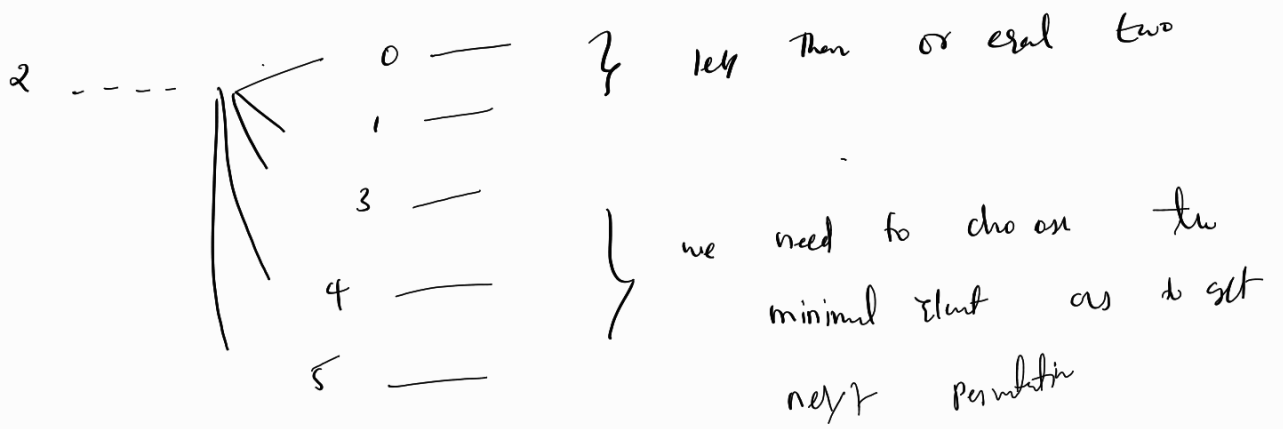
2 3 [→]   This rest of part can be drawn
            out by sorting or reversing
                the part.

At this point,
the point fell.
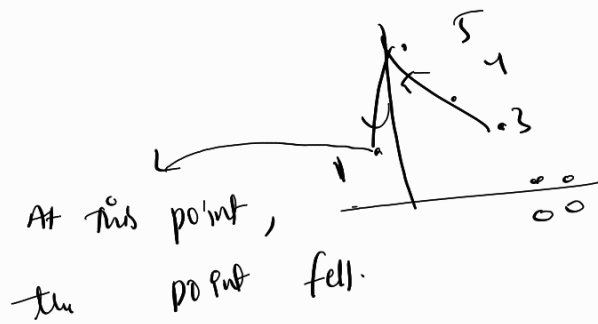
⇒ we can say that clearly
the elements are greater then
the before elements.

5 > 4 , 4 > 3 , 5 > 0

∴ we can   so as we got 1
swap these elements.
                             ie, 1 < 5
                                   ✓

from right side   step → ① Find index at which the array is
a[i] < a[i+1]   ←   pivoted.
                     ② Check if array is sorted based on
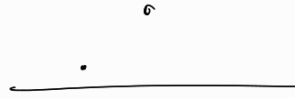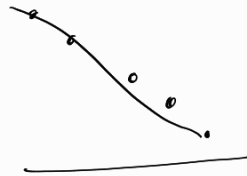   index
                     ③ check the min of nums greater then
                        arg(index) value .

if   index == -1 :

④   reverse the array

[1, 2, 3, 4 | 5]   =)   index fout

[5, 4, 3, 2, 1]

No   index   =)   -1.

so reverse it

⇒ as it's the lexiographically

the biggest one.