# «РГР» Increase Solved, Decrease Left

## The work of student Dmytro Kyba of KC-231 group

**Task 1**. Insertion sort.

Manually execute insertion sort algorithm. Calculate total number of moves and total number of insertions (except trivial “insertions” which actually do nothing).

Case number 18.

**(18).** 44 77 6 0 33 66 8 88 99 11 4 22 2 55;

Consider case 0 (array 44, 77, 6, 0, 33, 66, 8, 88, 99, 11, 4, 22, 2, 55).

**Main loop iteration i=1:**

- Array before: 44, 77, 6, 0, 33, 66, 8, 88, 99, 11, 4, 22, 2, 55
- Current element: 77
- No movement required (since 77>44), so the array remains unchanged.

**Array after**: 44, 77, 6, 0, 33, 66, 8, 88, 99, 11, 4, 22, 2, 55

**Main loop iteration i=2:**

- Array before: 44, 77, 6, 0, 33, 66, 8, 88, 99, 11, 4, 22, 2, 55
- Current element: 6
- Move elements 77,44 to the right to make space.
- Moves: 2
- Insert 6 at the beginning.

**Array after**: 6, 44, 77, 0, 33, 66, 8, 88, 99, 11, 4, 22, 2, 55

**Main loop iteration i=3:**

- Array before: 6, 44, 77, 0, 33, 66, 8, 88, 99, 11, 4, 22, 2, 55
- Current element: 0
- Move elements 77, 44, 6 to the right.
- Moves: 3
- Insert 0 at the beginning.

**Array after**: 0, 6, 44, 77, 33, 66, 8, 88, 99, 11, 4, 22, 2, 55

**Main loop iteration i=4:**

- Array before: 0, 6, 44, 77, 33, 66, 8, 88, 99, 11, 4, 22, 2, 55

- Current element: 33
- Move elements 77,44 to the right.
- Moves: 2
- Insert 33 in the current position.

**Array after**: 0, 6, 33, 44, 77, 66, 8, 88, 99, 11, 4, 22, 2, 55

**Main loop iteration i=5**

- Array before: 0, 6, 33, 44, 77, 66, 8, 88, 99, 11, 4, 22, 2, 55
- Current element: 66
- Move element 77 to the right.
- Moves: 1
- Insert 66 in the correct position.

**Array after**: 0, 6, 33, 44, 66, 77, 8, 88, 99, 11, 4, 22, 2, 55

**Main loop iteration i=6:**

- Array before: 0, 6, 33, 44, 66, 77, 8, 88, 99, 11, 4, 22, 2, 55
- Current element: 8
- Move elements 77, 66, 44, 33 to the right.
- Moves: 4
- Insert 8 in the correct position.

**Array after**: 0, 6, 8, 33, 44, 66, 77, 88, 99, 11, 4, 22, 2, 55

**Main loop iteration i=7:**

- Array before: 0, 6, 8, 33, 44, 66, 77, 88, 99, 11, 4, 22, 2, 55
- Current element: 88
- No movement required (since 88>77), so the array remains unchanged.

**Array after**: 0, 6, 8, 33, 44, 66, 77, 88, 99, 11, 4, 22, 2, 55

**Main loop iteration i=8:**

- Array before: 0, 6, 8, 33, 44, 66, 77, 88, 99, 11, 4, 22, 2, 55
- Current element: 99
- No movement required (since 99>88), so the array remains unchanged.

**Array after**: 0, 6, 8, 33, 44, 66, 77, 88, 99, 11, 4, 22, 2, 55

**Main loop iteration i=9:**

- Array before: 0, 6, 8, 33, 44, 66, 77, 88, 99, 11, 4, 22, 2, 55
- Current element: 11
- Move elements 99, 88, 77, 66, 44, 33 to the right.
- Moves: 6
- Insert 11 in the correct position.

**Array after**: 0, 6, 8, 11, 33, 44, 66, 77, 88, 99, 4, 22, 2, 55

**Main loop iteration i=10:**

- Array before: 0, 6, 8, 11, 33, 44, 66, 77, 88, 99, 4, 22, 2, 55
- Current element: 4
- Move elements 99, 88, 77, 66, 44, 33, 11, 8, 6 to the right.
- Moves: 9
- Insert 4 in the correct position.

**Array after**: 0, 4, 6, 8, 11, 33, 44, 66, 77, 88, 99, 22, 2, 55

**Main loop iteration i=11:**

- Array before: 0, 4, 6, 8, 11, 33, 44, 66, 77, 88, 99, 22, 2, 55
- Current element: 22
- Move elements 99,88,77,66,44,33 to the right.
- Moves: 6
- Insert 22 in the correct position.

**Array after**: 0, 4, 6, 8, 11, 22, 33, 44, 66, 77, 88, 99, 2, 55

**Main loop iteration i=12:**

- Array before: 0, 4, 6, 8, 11, 22, 33, 44, 66, 77, 88, 99, 2, 55
- Current element: 2
- Move elements 99, 88, 77, 66, 44, 33, 22, 11, 8, 6, 4 to the right.
- Moves: 11
- Insert 2 in the correct position.

**Array after**: 0, 2, 4, 6, 8, 11, 22, 33, 44, 66, 77, 88, 99, 55

**Main loop iteration i=13i = 13i=13**

- Array before: 0, 2, 4, 6, 8, 11, 22, 33, 44, 66, 77, 88, 99, 55
- Current element: 555555
- Move elements 99,88,77,6699, 88, 77, 6699,88,77,66 to the right.
- Moves: 4
- Insert 555555 in the correct position.

**Array after**: 0, 2, 4, 6, 8, 11, 22, 33, 44, 55, 66, 77, 88, 99

**Summary**

- Total moves: 48
- Total insertions: 10

**Task 2**. Exponentiation by Squaring.

Manually execute “Exponentiation by Squaring” algorithm for computing $(a^b) mod\ c$.

Write values of the algorithm’s variables at the very beginning, after each loop iteration and after each change of $res$. Explain with words at least $1/3$ of all steps; you may omit explanation, only when step is identical to some already explained one.

Use iterative version of exponentiation by squaring (not recursive one).

Case number 14.

(14). $(5860^{6802})\ mod\ 6331$

First of all, let’s fix the algorithm being used:

```
def mod_exp(a, m, c):
    b = a
    res = 1.0
    print(f"b = {b}, m = {m}, res = {res} (Before loop start)")

    while m > 0:
        if m % 2 == 1:
            res *= b
            res %= c
            print(f"b = {b}, m = {m}, res = {res} (Odd m, update res)")
        m //= 2
        b *= b
        b %= c
        print(f"b = {b}, m = {m}, res = {res}")

    return res
```

Manual calculation process:

| b | m | res | |
|---|---|---|---|
| 5860 | 6802 | 1 | Initial values before the loop starts. |
| 256 | 3401 | 1 | Value of $m$ was halved, and $\boldsymbol{b}$ was squared and taken modulo 6331, as $(5860 \times 5860) mod\ 6331 = 256$. |
| 256 | 3401 | 256 | $m$ is odd (3401), so $res$ is updated as $(1 \times 256)\ mod\ 6331 = 256$ |
| 2226 | 1700 | 256 | $m$ was halved, and $b$ was recalculated, $(256 \times 256)\ mod\ 6331 = 2226$ |
| 4234 | 850 | 256 | |
| 3695 | 425 | 256 | |
| 3695 | 425 | 2601 | |
| 3389 | 212 | 2601 | |
| 887 | 106 | 2601 | |
| 1725 | 53 | 2601 | |
| 1725 | 53 | 4377 | |
| 55 | 26 | 4377 | |
| 3025 | 13 | 4377 | |
| 3025 | 13 | 2304 | |
| 2330 | 6 | 2304 | |
| 3233 | 3 | 2304 | |
| 3233 | 3 | 3576 | |
| 6139 | 1 | 3576 | |
| 6139 | 1 | 3487 | $m$ is odd (1), so $res$ is updated as $(3576 \times 6139) \bmod 6331 = 3487$ |
| 5209 | 0 | 3487 | $m$ reached 0, ending the algorithm with the result $(5860^{6802})\ mod\ 6331 = 3487$ |

**Main result:** $(5860^{6802})\ mod\ 6331 = 3487.$