

## EGOI 2026 Editorial - Ferris Wheel

### Task Author: Nils Gustafsson

In this task, we are given  $N$  cabins numbered from 0 to  $N - 1$ . For each cabin, we know whether the next cabin on the wheel must have a larger number (+) or a smaller number (-). We must decide whether there exists a cyclic ordering of all cabins satisfying every requirement, and if so, print one such ordering.

*Key observation:* In a valid configuration: - cabin 0 can never have a - sign, because there exists no cabin with a smaller number; - cabin  $N - 1$  can never have a + sign, because there exists no cabin with a larger number.

### Subtask 1: $N = 3$

There are only three cabins: 0, 1, and 2.

Thanks to the *key observation*, if  $s_0 = -$  or  $s_2 = +$ , the answer is NO.

Otherwise, the answer is always YES. In this case, cabin 0 must have a + sign, cabin 2 must have a - sign, and cabin 1 can have either sign. If  $s_1 = +$ , we can print (0, 1, 2); otherwise, we can print (0, 2, 1).

### Subtask 2: Exactly one +

Again, cabin 0 cannot have a valid - sign. Since there is exactly one '+' sign, the only possible valid case is that the unique + sign is at position 0.

If this is not true, the answer is NO. Otherwise, one valid ordering is

$$0, N - 1, N - 2, \dots, 2, 1.$$

In this ordering, cabin 0 is followed by a larger number, and every other cabin is followed by a smaller number. The last cabin in the ordering is 1, and it is followed cyclically by 0, which is smaller. Thus, the construction satisfies all the conditions.

### Subtask 3: Alternating signs

In this subtask, the signs alternate.

For an alternating string that admits a valid configuration, the conditions from the *key observation* ( $s_0 = +$  and  $s_{N-1} = -$ ) determine the whole string. In fact, by examining the cases: - if  $N$  is odd, then  $s_0 = s_{N-1}$  and the answer is clearly NO (if they are both + then  $N - 1$  is not valid, otherwise 0 is not valid); - if

$N$  is even, then  $s_0 \neq s_{N-1}$  and the answer is YES if  $s_0 = +$  and NO otherwise (both  $N - 1$  and  $0$  are not valid).

In particular, in the only valid case,  $N$  must be even and the string must be of the form:

$$(+, -, +, -, +, -, \dots, +, -).$$

One valid solution is then to print the swapped adjacent pairs

$$(1, 0), (3, 2), (5, 4), \dots, (N - 3, N - 4),$$

and then finish with the pair  $(N - 2, N - 1)$ . Thus the full order is

$$1, 0, 3, 2, 5, 4, \dots, N - 3, N - 4, N - 2, N - 1.$$

Let us check the transitions. In each swapped pair  $(2k + 1, 2k)$ , the first cabin has a  $-$  sign, so going from  $2k + 1$  to  $2k$  is valid. The second cabin has a  $+$  sign, and the next printed cabin is larger, so the transition out of the pair is also valid. After the last swapped pair, cabin  $N - 4$  has a  $+$  sign and is followed by  $N - 2$ , which is larger. Then cabin  $N - 2$  has a  $+$  sign and is followed by  $N - 1$ , which is larger. Finally, cabin  $N - 1$  has a  $-$  sign and goes back cyclically to  $1$ , which is smaller.

This solves the alternating subtask in  $\mathcal{O}(N)$  time.

### Subtask 4 and Full Solution

We now prove that the two necessary conditions from the *key observation* are also sufficient: that is, if  $s_0 = +$  and  $s_{N-1} = -$ , then a valid Ferris wheel ordering exists.

Assume  $s_0 = +$  and  $s_{N-1} = -$ . Construct the answer like this:

1. Print all indices  $i$  with  $s_i = +$  in increasing order.
2. Then print all indices  $i$  with  $s_i = -$  in decreasing order.

For example, if

$$s = (+, -, +, -, -),$$

then the  $+$  cabins are  $(0, 2)$ , and the  $-$  cabins in decreasing order are  $(4, 3, 1)$ , so we may print

$$0, 2, 4, 3, 1.$$

Let us check why this always works. Notice that there is always at least one + cabin (0) and at least one - cabin ( $N - 1$ ).

In the first part, all + cabins except the last one are followed by larger + cabins. By construction, the last + cabin is followed by  $N - 1$ , which is strictly larger.

In the second part, all - cabins are printed in decreasing order, so each one except the last is followed by a smaller - cabin. Since we cyclically return to the first printed cabin at the end, the last - cabin is followed by 0, which is strictly smaller.

This proves both the construction and the characterization of the impossible cases.

For Subtask 4, we have  $N \leq 1000$ , so it suffices to maintain one list while iterating through the cabins and inserting each index either at the beginning or at the end. Since inserting at the beginning takes linear time, this implementation runs in  $\mathcal{O}(N^2)$ .

For the full solution, we need a faster implementation. One possibility is to store all the + cabins into one list, in increasing order, and all - cabins into another list, in decreasing order. Then print the + list followed by the - list. This can be done in  $\mathcal{O}(N)$  time, by first reading the string and storing the + cabins and then reading it backwards and storing the - cabins.

### Alternative full solution

There is also a different approach, built on Subtask 3. Every string can be seen as an alternating sequence of blocks of consecutive + or -.

For example, if

$$s = (+, +, +, -, -, +, -, +, +, -, -),$$

we split it into blocks as follows

$$(+, +, +), (-, -), (+), (-), (+, +), (-, -)$$

and thus group the corresponding cabins like this:

$$(0, 1, 2), (3, 4), (5), (6), (7, 8), (9, 10).$$

We know how to deal with alternating strings, and we can apply the ordering of Subtask 3 to the blocks. It only remains to sort the cabins inside each block.

- if the block is made of [+], we place the cabins inside it in increasing order;
- if the block is made of [-], we place the cabins inside it in decreasing order.

In the example above, first order the numbers according to the blocks in this way

$$(0, 1, 2), (4, 3), (5), (6), (7, 8), (10, 9).$$

and then applying the construction of Subtask 3 to the alternating blocks gives:

$$(4, 3), (0, 1, 2), (6), (5), (7, 8), (10, 9).$$

This gives us the following construction:

$$4, 3, 0, 1, 2, 6, 5, 7, 8, 10, 9.$$

It can be checked, similarly to the other constructions, that this always works. This construction can also be done in  $\mathcal{O}(N)$  time.