Bestsellers II

Submission deadline:	**2022-11-22 23:59:59**
Late submission with malus:	**2023-02-01 23:59:59** (Late submission malus: 100.0000 %)
Evaluation:	**12.0120**
Max. assessment:	**1.0000** (Without bonus points)
Submissions:	21 / -
Advices:	0 / 0

I have traveled the world far and wide, and I can honestly tell that one cannot find any other nation that would enjoy Christmas as much as the Whos. As I have learned from my many encounters with the Whos, it is not only the idea of gifts, merrymaking in the snow and time spent together with family and friends. The feelings that the Whos hold towards Christmas come from deep within their hearts and I dare to call it an animal desire or an instinctual need for joyous and peaceful Christmas.

Or at least for the majority of them. As in any other group, there are exceptions to the rule among the Whos. And this one time, I don't mean the Grinch. I have somebody even worse on my mind. What can be worse than someone trying to steal Christmas out of his own fear and feeling of injustice, you ask? Well, one who uses the deepest human desires, nourishes fear and insecurity and claims to have an easy solution to every problem. And all that with a singular goal in mind – to make a fortune. Yes, dear reader, you guessed correctly. I am talking about none other than the honorable mayor of Whoville.

In the case of such a naïve nation as the Whos, it comes as no surprise that the mayor governs Whoville for the umpteenth term in a row, always gaining more than 99 % of votes. The more experienced of us are surely not taken aback by the fact that the town's well-being is merely an afterthought for him. The thing is that the mayor owns the largest toy store in Whoville (and the only one if I'm being honest). A colossal conflict of interest, one would say! But no one in Whoville as much as raises an eyebrow.

As they say, appetite comes with eating, and the mayor craves more. Hence, he initiated a campaign last year with a provisional title Bestsellers. It involved a massive digitalization of all the purchases and analysis of products' merchantability. Encouraged by its enormous success that increased his profits tenfold, the mayor intends to continue this year and push digitalization even further. Ladies and gentlemen, please welcome Bestsellers II, an even more monstrous campaign designed to utterly drain the wallets of Whoville. Some people even say that the campaign will take away not only the Whos' savings, but also small pieces of their souls.

And how do you fit into the picture, you ask? You are the good guy, surely you won't get your hands dirty with such a wicked affair. However, the sad truth is that no one other than the town board and the mayor's toy store will offer you any contract. Plus, you proved yourself very capable last year. That is why you shall implement Bestsellers this year again.

Interface

Your task is to implement the template `Bestsellers` parameterized by the `Product` type which is used as an identification of individual products. The `Product` type has a copy constructor (and a copy assignment operator), destructor, comparison operators and a specialization of `std::hash`. The class `Bestsellers` must implement the following public methods:

- `size_t products() const` returns the number of different product types on record (**not** the number of pieces sold),
- `void sell(const Product& p, size_t amount)` records selling `amount` pieces of product `p`,
- `size_t rank(const Product& p) const` returns the position of `p` in the list of all recorded product types sorted by pieces sold in descending order with indexing starting at 1 (`rank(p) = 1` for the best-selling product `p`); if there are more product types with the same number of pieces sold, you can order them arbitrarily,
- `const Product& product(size_t rank) const` inverse function to `rank` (unless an exception occurs, it holds that `product(rank(p)) == p` and `rank(product(r)) == r`),
- `size_t sold(size_t r) const` number of pieces sold of the `r`-th best-selling product,
- `size_t sold(size_t from, size_t to) const` analogous to the single-parameter version, it returns the sum of pieces sold over the interval of ranks from `from` to `to` (including both endpoints; i.e., `sold(r, r) == sold(r)` and `sold(1, products())` is the total number of pieces sold over all products; input `to < from` is considered invalid).
- **Bonus only:** `size_t first_same(size_t r) const` and `size_t last_same(size_t r) const` returns the smallest (resp. largest) rank `r'` such that `sold(r') == sold(r)`. These methods are used only in the bonus test. If you are not solving the bonus test, leave the default implementation that always returns 0.

If any argument is invalid (i.e. rank is out of range or an unknown product is input to `rank`), raise the exception `std::out_of_range`.

Classification Conditions

- To obtain 1 point, you must implement the interface described above, but the implementation does not need to be too efficient.
- To obtain 3.5 points, the implementation must be efficient if the number of pieces sold are random, but the complexity of the method `sold(from, to)` can depend on `to - from`.

- To obtain 7 points, the implementation must be efficient, the complexity of the method `sold(from, to)` can still depend on `to - from`.
- To obtain 10 points, the method `sold(from, to)` must be efficient even for large `to - from`.
- **Bonus:** To obtain 12 points, you must correctly and efficiently implement the methods `first_same` and `last_same`.

Notes

- Think through what is necessary to pass individual tests.
- Solve tests progressively. Do not try to obtain all points, including the bonus, on your first submit.
- When improving your implementation, you can use older and slower versions for testing on large inputs.
- `Product` type always has a default constructor. However, anyone who passed PA2 should not use it.

Sample data: **Download**

Reference

- **Evaluator: computer**
  - Program compiled
  - Test 'Základní test (s mem-debuggerem)': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.071 s (limit: 12.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Náhodný test bez velkých intervalových dotazů': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.920 s (limit: 10.000 s)
    - Bonus test - success, evaluation: 350.00 %
  - Test 'Velký test bez velkých intervalových dotazů': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.935 s (limit: 10.000 s)
    - Bonus test - success, evaluation: 200.00 %
  - Test 'Velký test': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 1.153 s (limit: 10.000 s)
    - Bonus test - success, evaluation: 143.00 %
  - Test 'Bonus test': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.629 s (limit: 10.000 s)
    - Bonus test - success, evaluation: 120.00 %
  - Overall ratio: 1201.20 % (= 1.00 * 3.50 * 2.00 * 1.43 * 1.20)
- Total percent: 1201.20 %
- Early submission bonus: 0.25
- Total points: 12.01 * (1.00 + 0.25) = 15.01

SW metrics:

	Total	Average	Maximum	Function name
Functions:	**31**	--	--	--
Lines of code:	**146**	**4.71 ± 3.49**	**14**	`Bestsellers::Tree::remove_node`
Cyclomatic complexity:	**72**	**2.32 ± 1.45**	**6**	`Bestsellers::Tree::balance_and_update`

21 **2022-11-22 18:08:45** **Download**

Submission status: Evaluated
Evaluation: 12.0120

- **Evaluator: computer**
  - Program compiled
  - Test 'Basic test (with mem-debugger)': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.075 s (limit: 12.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Random data test without large interval queries': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 1.438 s (limit: 10.000 s)
    - Bonus test - success, evaluation: 350.00 %
  - Test 'Big test without large interval queries': success

        - result: 100.00 %, required: 100.00 %
        - Total run time: 2.266 s (limit: 10.000 s)
        - Bonus test - success, evaluation: 200.00 %
    - Test 'Big test': success
        - result: 100.00 %, required: 100.00 %
        - Total run time: 2.195 s (limit: 10.000 s)
        - Bonus test - success, evaluation: 143.00 %
    - Test 'Bonus test': success
        - result: 100.00 %, required: 100.00 %
        - Total run time: 3.731 s (limit: 10.000 s)
        - Bonus test - success, evaluation: 120.00 %
    - Overall ratio: 1201.20 % (= 1.00 * 3.50 * 2.00 * 1.43 * 1.20)
- Total percent: 1201.20 %
- Total points: 12.01 * 1.00 = 12.01

SW metrics:

	Total	Average	Maximum	Function name
Functions:	**41**	--	--	--
Lines of code:	**530**	**12.93 ± 13.33**	67	`delete_product_from_tree`
Cyclomatic complexity:	**187**	**4.56 ± 4.51**	22	`execute_tree_balancer`

20	**2022-11-22 18:05:34**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
    - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

19	**2022-11-22 18:00:14**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
    - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

18	**2022-11-22 17:58:58**	**Download**
Submission status:	Evaluated	
Evaluation:	10.0100	

- **Evaluator: computer**
    - Program compiled
    - Test 'Basic test (with mem-debugger)': success
        - result: 100.00 %, required: 100.00 %
        - Total run time: 0.076 s (limit: 12.000 s)
        - Mandatory test success, evaluation: 100.00 %
    - Test 'Random data test without large interval queries': success
        - result: 100.00 %, required: 100.00 %
        - Total run time: 1.332 s (limit: 10.000 s)
        - Bonus test - success, evaluation: 350.00 %
    - Test 'Big test without large interval queries': success
        - result: 100.00 %, required: 100.00 %
        - Total run time: 1.992 s (limit: 10.000 s)
        - Bonus test - success, evaluation: 200.00 %
    - Test 'Big test': success
        - result: 100.00 %, required: 100.00 %
        - Total run time: 2.224 s (limit: 10.000 s)
        - Bonus test - success, evaluation: 143.00 %
    - Test 'Bonus test': failed
        - result: 21.09 %, required: 100.00 %
        - Total run time: 3.494 s (limit: 10.000 s)
        - Bonus test - failed, evaluation: No bonus awarded
        - Failed (invalid output)
        - Failed (invalid output)
        - Failed (invalid output)
        - Failed (invalid output)

        - Failed (invalid output)
        - Failed (invalid output)
        - Failed (invalid output)
        - Failed (invalid output)
    - Overall ratio: 1001.00 % (= 1.00 * 3.50 * 2.00 * 1.43)
- Total percent: 1001.00 %
- Total points: 10.01 * 1.00 = 10.01

SW metrics:

	Total	Average	Maximum	Function name
Functions:	**41**	--	--	--
Lines of code:	**520**	**12.68 ± 13.50**	67	`delete_product_from_tree`
Cyclomatic complexity:	**183**	**4.46 ± 4.56**	22	`execute_tree_balancer`

17	**2022-11-22 16:50:44**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
    - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

16	**2022-11-22 16:31:21**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
    - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

15	**2022-11-22 16:22:36**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
    - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

14	**2022-11-22 16:07:33**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
    - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

13	**2022-11-22 15:35:31**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
    - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

12	**2022-11-22 15:34:25**	**Download**
Submission status:	Evaluated	

Evaluation:	0.0000

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

11	**2022-11-22 15:15:06**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

10	**2022-11-22 15:12:22**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

9	**2022-11-22 15:05:48**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

8	**2022-11-22 15:03:25**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

7	**2022-11-22 15:02:40**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

6	**2022-11-22 15:00:43**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

5	2022-11-22 14:54:55	Download
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

4	2022-11-20 03:54:53	Download
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

3	2022-11-20 03:49:57	Download
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

2	2022-11-20 03:48:22	Download
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00

1	2022-11-20 03:31:30	Download
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Compile in 'basic' mode failed.
- Total percent: 0.00 %
- Total points: 0.00 * 1.00 = 0.00