

## Zadanie 5: Poprzednie w zakresie

**Opened:** Monday, 18 December 2023, 10:00 AM

**Due:** Wednesday, 17 January 2024, 4:05 PM

Twoim zadaniem jest zaimplementowanie struktury danych umożliwiającej efektywne odpowiadanie na określone zapytania dotyczące ciągu liczb  $X$ . Co więcej, kolejne elementy ciągu mogą być ujawniane on-line.

Dla ustalonego ciągu  $X = x_0, x_1, \dots, x_{n-1}$  interesuje nas funkcja:

$\text{prevInRange}(i, [lo, hi])$

która wyznacza **największy** indeks  $j$  taki, że  $0 \leq j \leq i$  oraz  $x_j \in [lo, hi]$ . Jeśli taki indeks nie istnieje, wynikiem funkcji powinno być -1.

Czyli:

$\text{prevInRange}(i, [lo, hi]) = \max\{0 \leq j \leq i : x_j \in [lo, hi]\}$  lub -1 (jeśli taki indeks nie istnieje)

Twój moduł powinien udostępniać następujące funkcje:

`void init(const vector<int> &x)` – inicjalizacja początkowego ciągu  $X$  przez wartości wektora  $x$  (uwaga: ciąg może zawierać dowolne wartości mieszczące się w typie `int`)

`int prevInRange(int i, int lo, int hi)` – oblicz wartość  $\text{prevInRange}(i, [lo, hi])$ . Możesz założyć, że  $0 \leq i < |X|$  i  $\text{INT\_MIN} \leq lo \leq hi \leq \text{INT\_MAX}$ .

`void pushBack(int v)` – dodanie na końcu obecnego ciągu  $X$  elementu o wartości  $v$

`void done()` – zwolnienie całej pamięci używanej do obsługi ciągu  $X$ .

Deklaracje podanych funkcji znajdują się w pliku [prev.h](#). Twoim zadaniem jest zaimplementowanie podanych funkcji w pliku `prev.cpp`.

Przykład interakcji znajduje się w załączonym pliku [main.cpp](#).

Komenda kompilacji:

```
g++ @opcjeCxx main.cpp prev.cpp -o main.e
```

Aby Twoje rozwiązanie uzyskało maksymalną punktację, (**zamortyzowany**) koszt czasowy funkcji `prevInRange` oraz `pushBack` musi wynosić  $O(\log z)$ , a złożoność funkcji `init` powinna być w najgorszym razie  $O(|X| \log z)$ , przy czym  $z$  to **zakres typu** `int`. Za dodatkowy logarytm w złożoności można stracić 1 punkt. Rozwiązanie siłowe dostanie 0, sorry.


Twoje rozwiązanie zostanie także uruchomione za pomocą narzędzia `valgrind`, które pozwala sprawdzać m.in., czy program nie miał wycieków pamięci. W przypadku wykrycia wycieków pamięci za pomocą komendy:

```
valgrind --tool=memcheck --leak-check=yes ./main.e
```


możesz stracić od 1 do 2 punktów za zadanie.

**Uwaga:** Tym razem na laboratorium omówiona zostanie struktura danych, której można użyć do rozwiązania zadania. Twoje zadanie będzie polegało na zaimplementowaniu jej. W implementacji można używać `shared_ptr`.

Submission status

|                     |   |
|---------------------|---|
| Attempt number      | This is attempt 1.  |
| Submission status   | Submitted for grading   |
| Grading status      | Graded  |
| Time remaining      | Assignment was submitted 2 days 22 hours early  |
| Last modified       | Sunday, 14 January 2024, 5:54 PM  |
| File submissions    | <div><div> <a href="#">prev.cpp</a></div><div>14 January 2024, 5:54 PM</div></div> |
| Submission comments | <div><div></div><div><a href="#">Comments (0)</a></div></div>   |

Feedback

|           |  |
|-----------|--|
| Grade     | 5.00 / 5.00  |
| Graded on | Monday, 15 January 2024, 10:12 AM  |
| Graded by | <div><div></div><div>Łukasz Bożyk</div></div> |

Contact us



Follow us

 Contact site support

You are logged in as Witold Formański (Log out)

Data retention summary

Get the mobile app

Get the mobile app

This theme was developed by

conecti.me

Moodle, 4.1.10 (Build: 20240422) | [moodle@mimuw.edu.pl](mailto:moodle@mimuw.edu.pl)