

Scenariusz zajęć nr 10

Temat: Biblioteka

Cele:

W trakcie zajęć uczeń zapozna się z następującymi pojęciami:

- Szybkie wczytywanie i wypisywanie dużej ilości danych,
- Wykonywanie dużej ilości zapytań (ang. *queries*) dotyczących określenia położenia lub dokładania książek,
- Używanie pomocniczej tablicy o wartościach całkowitych zawierającej informacje o położeniu książek,
- Użycie operatorów ++ oraz -- w instrukcjach podstawienia.

Wstęp:

Uruchomienie środowiska programistycznego, zapoznanie się z edytorem kodu źródłowego i sposobem kompilacji programu (kontynuacja).

Dla nauczyciela:

W razie braku zainstalowanego środowiska można skorzystać z dowolnego środowiska dostępnego w sieci Internet, na przykład `cpp.sh`.

Przebieg zajęć:

Zapoznanie się z treścią zadania:

Treść zadania jest dołączona do scenariusza zajęć w formie pliku `cpp_10-biblioteka.pdf`.

Dla nauczyciela:

Tematem zajęć jest zadanie *Biblioteka* dostępne na serwisie `szkopul.edu.pl`, wzorowane na zadaniu *Books Queries* (1066C) z serwisu `codeforces.com`. Zadanie można rozwiązać w dowolnym języku programowania, a jego stopień trudności należy ocenić jako właściwy dla uczestników średnio zaawansowanego kursu programowania.

Wskazówki do rozwiązania zadania:

W tym zadaniu główną trudnością jest szybkie udzielenie odpowiedzi na pytanie, ile książek znajduje się po lewej i po prawej stronie danej książki, której *id* występuje w zapytaniu. W tym celu warto ponumerować książki stojące na półce tak, aby wiadomo było, które miejsce zajmuje. Problem polega na tym, że numer taki nie może być liczbą naturalną, bo książki dokładamy i z prawej, i z lewej strony. Musi to więc być liczba całkowita, a numer zero powinien wypadać z grubsza gdzieś w środku półki.

Na przykład, jeśli na półce stoją w danym momencie książki o numerach:

$$-3, -2, -1, 0, 1, 2, 3, 4$$

to dokładając książkę z lewej strony nadamy jej numer -4 , a z prawej: 5 .

Do tego potrzebne są dwie zmienne oznaczające skrajne numery książek: *lewo* oraz *prawo* – jedna z nich jest uaktualniana przy dokładaniu kolejnej książki.

Nadany numer zapisujemy w pomocniczej tablicy *a* jako $a[id]$. Zakres identyfikatorów książek pozwala na umieszczenie takiej tablicy w programie.

Wartości początkowe zmiennych *lewo* i *prawo* powinny być w okolicy zera, na przykład $lewo = 1$, $prawo = 0$. Należy tylko zadbać, aby numer 0 otrzymała tylko jedna książka.

Jeśli zapytanie dotyczy książki o identyfikatorze *id*, to na lewo od niej stoi $a[id] - lewo$ książek, zaś na prawo: $prawo - a[id]$. Wynikiem zapytania jest mniejsza z tych liczb.

Kod programu w języku C++:

```
#include <iostream>
#include <algorithm>
using namespace std;

int a[200001];

int main(){
    ios_base::sync_with_stdio(0);
    int q, b, lewo = 1, prawo = 0;
    char c;
    cin >> q;
```

```
for(int i = 0; i < q; i++)
{
    cin >> c >> b;
    if(c == 'L')
        a[b] = --lewo;
    else if(c == 'R')
        a[b] = ++prawo;
    else
        cout << min(a[b] - lewo, prawo - a[b]) << endl;
}
return 0;
}
```

Podsumowanie i uwagi dodatkowe:

Instrukcja `ios_base::sync_with_stdio(0)` powoduje wyłączenie synchronizacji pomiędzy biblioteką `iostream` i alternatywną biblioteką wejścia/wyjścia `cstdio`. Ta druga biblioteka pochodzi jeszcze z języka ANSI C. Jeśli jej nie używamy (w szczególności jeśli nie używamy funkcji `scanf/printf`), raczej powinniśmy wyłączyć synchronizację, co zaowocuje znaczącym przyspieszeniem operacji wejścia/wyjścia.