
PPPD - Lab. 07

Copyright ©2021 M. Śleszyńska-Nowak i in.

Zadanie punktowane, lab 07, 2019/2020, autor: Maciej Bartoszek

Temat: Multizbiór

Treść zadania

W zadaniu nie można korzystać z `append`, `slice` i indeksowania ujemnego.

Wstęp

Zadanie polega na zaimplementowaniu multizbioru. Multizbiór będziemy przechowywać jako listę, gdzie na i -tej pozycji przechowywana jest liczność liczby i . I tak, lista `[0, 1, 2, 0, 3]` oznacza multizbiór $\{1, 2, 2, 4, 4, 4\}$.

Przy wszystkich podpunktach możemy założyć, że:

- listy wejściowe spełniają regułę, że są najmniejszymi możliwymi reprezentacjami multizbiorów. Innymi słowy, ostatni element każdej listy jest dodatni,
- zbiór wynikowy będzie niepusty,
- elementy multizbioru są nieujemne.

Zadanie

1. Napisz funkcję `podaj()`, która pyta użytkownika o maksymalny element przechowywany w multizbiorze, a następnie prosi kolejno o podanie elementu i -tego listy reprezentującej multizbiór (jeśli podamy np. 5 na pozycji 3, to oczywiście znaczy, że chcemy przechowywać pięć trójek). Funkcja zwraca tę listę. Jeśli ostatni element listy jest równy zero, rzucamy wyjątek (bo lista nie spełnia warunków zadania). W `main()` wczytaj multizbiór.

2. Napisz funkcję `wypisz(lista)`, która wypisuje na konsolę multizbiór. Przykładowo:

```
wypisz([0,1,2,0,3])  
[1,2,2,4,4,4]
```

3. Napisz funkcję `dodaj(zbior, element)`, która dodaje element do zbioru. Jeśli element jest z zakresu $\{0, 1, \dots, \text{rozmiar} - 1\}$, to powinniśmy zwracać tę samą listę, którą dostaliśmy jako argument (oczywiście zmodyfikowaną). Jeśli element wychodzi poza rozmiar listy, to stwórz nową listę o odpowiednim rozmiarze i przepisz stare elementy.

W `main()` wczytaj element od użytkownika i dodaj go do zbioru.

Przykład: dostaliśmy multizbiór reprezentowany przez `[0, 2]`, a mamy dodać tam jeszcze jedną jedynkę 1. Powinniśmy zmodyfikować argument wejściowy i go zwrócić w ten sposób, że wynik to `[0, 3]`. Przykład 2: dostaliśmy multizbiór reprezentowany przez `[0, 2]`, a mamy dodać tam trójkę. Wtedy musimy stworzyć nową listę i przepisać do niej stare elementy, oraz dodać trójkę: `[0, 2, 0, 1]`. Argument wejściowy pozostaje niezmieniony.

-
4. Napisz funkcję `przeciecie(zbiorA, zbiorB)`, która oblicza przecięcie dwóch zbiorów i zwraca je w postaci listy. Co więcej, lista przechowująca zbiór wynikowy powinna być najmniejsza możliwa (czyli nie może być zer na końcu, ostatnim elementem powinna być wartość ostro większa od zera).

W `main()` wczytaj kolejny multizbiór i wypisz przecięcie dwóch multizbiorów. Wypisz wynik na ekran.

Przykład 1: wynikiem przecięcia zbiorów przechowywanych za pomocą list $A = [0, 3, 0, 4, 0, 1]$ i $B = [1, 2, 2, 4, 1]$ powinno być $[0, 2, 0, 4]$. Zmniejszona lista wynikowa wynika z faktu, że liczba 3 jest największą liczbą powtarzającą się w obu zbiorach. Stąd długość równa 4. Przykład 2: Dla list $A = [1, 2]$ i $B = [2, 1]$ wynikiem jest $[1, 1]$ - bo powtarza się jedno zero i jedna jedynka w obu zbiorach.

5. Napisz funkcję `roznica(zbiorA, zbiorB)`, która oblicza różnicę dwóch zbiorów. Co więcej, lista przechowująca zbiór wynikowy powinna być najmniejsza możliwa.

W `main()` odejmij jeden wczytany multizbiór od drugiego. Wypisz wynik na ekran.

Przykład 1: Wynikiem różnicy zbiorów przechowywanych za pomocą list $A = [0, 3, 0, 4]$ i $B = [1, 2, 2, 4]$ powinno być $A \setminus B = [0, 1]$. Zmniejszony rozmiar wynika stąd, że nie ma dwójek w zbiorze A , a wszystkie trójki odjęliśmy (bo było ich tyle samo w zbiorze B). Przykład 2: Dla list $A = [1, 2]$ i $B = [2, 1]$ wynikiem jest $[0, 1]$.

Przykład 1:

```
Podaj rozmiar zbioru A
4
Podaj elementy zbioru A:
Podaj ilość 0: 0
Podaj ilość 1: 3
Podaj ilość 2: 0
Podaj ilość 3: 3
Podany zbiór to:
[1,1,1,3,3,3]
Podaj nowy element zbioru A
3
Zbiór A po dodaniu elementu:
[1,1,1,3,3,3,3]
Podaj rozmiar zbioru B
4
Podaj elementy zbioru B:
Podaj ilość 0: 1
Podaj ilość 1: 2
Podaj ilość 2: 2
Podaj ilość 3: 4
Roznica zbiorow:
[1]
Przeciecie zbiorow:
[1,1,3,3,3,3]
```

Przykład 2:

```
Podaj rozmiar zbioru A
4
```

Podaj elementy zbioru A:
Podaj ilość 0: 0
Podaj ilość 1: 3
Podaj ilość 2: 0
Podaj ilość 3: 4
Podany zbiór to:
[1,1,1,3,3,3,3]
Podaj nowy element zbioru A
5
Zbiór A po dodaniu elementu:
[1,1,1,3,3,3,5]
Podaj rozmiar zbioru B
5
Podaj elementy zbioru B:
Podaj ilość 0: 1
Podaj ilość 1: 2
Podaj ilość 2: 2
Podaj ilość 3: 4
Podaj ilość 4: 1
Różnica zbiorów:
[1, 5]
Przecięcie zbiorów:
[1,1,3,3,3,3]

Punktacja

Za poszczególne etapy można uzyskać następującą liczbę punktów:

- Każdy podpunkt za 2 pkt.

Uwaga

- Jeśli program się nie kompiluje (interpretuje), ocena jest zmniejszana o połowę.
- Jeśli kod programu jest niskiej jakości (nieestetycznie formatowanie, mylące nazwy zmiennych itp.), ocena jest zmniejszana o 2 p.