

## Struktury danych i złożoność obliczeniowa

## Laboratorium 6 – Tablica mieszająca

Celem zadania jest implementacja słownika w postaci tablicy mieszającej.

Pojedynczy obiekt danych dla słownika zawiera tylko jedną składową - klucz typu całkowitego (**int**).

W implementacji słownika należy zastosować metodę adresowania otwartego, przy czym do rozwiązywania problemu kolizji możliwe jest zastosowanie dwóch różnych metod: adresowania liniowego i „klasycznego” mieszania podwójnego.

Pierwotna (główna) funkcja mieszająca ma postać:

$$H(\text{key}) = ((\text{key} \bmod 1000) + 2^{(\text{key} \bmod 10)} + 1) \bmod 997$$

Np.:  $H(20709) = ((20709 \bmod 1000) + 2^{(20709 \bmod 10)} + 1) \bmod 997 =$   
 $= (709 + 2^9 + 1) \bmod 997 = (709 + 512 + 1) \bmod 997 = 1222 \bmod 997 = 225$

Dodatkowa funkcja mieszająca wykorzystywana do rozwiązywania problemu kolizji w metodzie mieszania podwójnego ma postać:

$$H'(\text{key}) = (3 * \text{key}) \bmod 19 + 1$$

Np.:  $H'(20709) = (3 * 20709) \bmod 19 + 1 = 62127 \bmod 19 + 1 = 17$

Słownik powinien być zaimplementowany tak, by można było w nim umieścić co najmniej **997** obiektów danych.

Należy stworzyć funkcje realizujące:

- inicjację pustej tablicy mieszającej (przyjąć, że wartość początkowa dla pustej, czyli nie zawierającej obiektu danych, pozycji w tablicy mieszającej wynosi **0**);  
*argumenty: opcjonalnie wskaźnik/nazwa tablicy;*
- usunięcie z tablicy obiektu danych (klucza) o podanej jako argument wartości (wraz z obsługą przypadku, w którym obiekt o danym kluczu nie istnieje; przyjąć, że w przypadku odnalezienia i usunięcia wskazanego obiektu (klucza) wartość pozycji w tablicy mieszającej, z której usunięto obiekt, będzie wynosiła **-1**);  
*argumenty: opcjonalnie wskaźnik/nazwa tablicy, obowiązkowo klucz;*
- wstawienie do tablicy nowego obiektu danych o zadany klucz;  
*argumenty: opcjonalnie wskaźnik/nazwa tablicy, obowiązkowo klucz;*
- wstawienie do tablicy **X** nowych obiektów danych (kluczy) wygenerowanych losowo z zakresu od **20000** do **40000** (**UWAGA! Jeżeli w tablicy już znajduje się obiekt danych o wartości równej wylosowanej do wstawienia, to należy powtarzać losowanie; po zakończeniu działania tej funkcji w tablicy powinno się pojawić dokładnie X nowych wartości kluczy**);  
*argumenty: opcjonalnie wskaźnik/nazwa tablicy, obowiązkowo wartość X;*
- wyszukanie w tablicy obiektu danych (klucza) o podanej jako argument wartości (wraz z obsługą przypadku, w którym obiekt o danym kluczu nie istnieje);  
*argumenty: opcjonalnie wskaźnik/nazwa tablicy, obowiązkowo klucz;*
- wyświetlanie wartości z obszaru tablicy określonego przez indeks początkowy i końcowy tego obszaru;  
*argumenty: opcjonalnie wskaźnik/nazwa tablicy, obowiązkowo indeks początkowy i końcowy wyświetlanego obszaru.*

Tablicę mieszającą można zaimplementować strukturalnie (C) lub obiektowo(C++). W przypadku implementacji strukturalnej wskaźnik/nazwa tablicy powinna zostać przekazana do implementowanych funkcji jako argument.

Program po uruchomieniu wczytuje plik wejściowy `inlab06.txt`.

Plik `inlab06.txt` zawiera w pierwszej linii kolejno liczbę elementów do wylosowania **X**, a następnie wartości czterech kluczy **k1**, **k2**, **k3**, **k4**.

Następnie wywoływana jest sekwencja funkcji (dalej w funkcji `main()`):

- czas start;
- „ustal” jako mechanizm rozwiązywania kolizji adresowanie liniowe;
- zainicjuj „pustą” tablicę mieszającą
- usuń obiekt o wartości klucza **k1**;
- wstaw obiekt o wartości klucza **k1**;
- wyświetl obszar tablicy o indeksach od **0** do **100**;
- wstaw **X** obiektów do tablicy;
- wyświetl obszar tablicy o indeksach od **0** do **100**;
- wstaw kolejno obiekty o wartościach klucza **k2**, **k3** i **k4**;
- wyświetl obszar tablicy o indeksach od **0** do **100**;
- wyświetl obszar tablicy o indeksach od **500** do **600**;
- usuń obiekt o wartości klucza **k3**;
- usuń obiekt o wartości klucza **k4**;
- wyświetl obszar tablicy o indeksach od **0** do **100**;
- wyświetl obszar tablicy o indeksach od **500** do **600**;
- czas stop;
- wypisz czas wykonania tej fazy programu;
- czas start;
- „ustal” jako mechanizm rozwiązywania kolizji mieszanie podwójne;
- ponownie zainicjuj „pustą” tablicę mieszającą
- usuń obiekt o wartości klucza **k1**;
- wstaw obiekt o wartości klucza **k1**;
- wyświetl obszar tablicy o indeksach od **0** do **100**;
- wstaw **X** obiektów do tablicy;
- wyświetl obszar tablicy o indeksach od **0** do **100**;
- wstaw kolejno obiekty o wartościach klucza **k2**, **k3** i **k4**;
- wyświetl obszar tablicy o indeksach od **0** do **100**;
- wyświetl obszar tablicy o indeksach od **500** do **600**;
- usuń obiekt o wartości klucza **k3**;
- usuń obiekt o wartości klucza **k4**;
- wyświetl obszar tablicy o indeksach od **0** do **100**;
- wyświetl obszar tablicy o indeksach od **500** do **600**;
- czas stop;
- wypisz czas wykonania.

## Przygotowanie e-maila do wysłania:

**Uwaga!** Kod źródłowy programu (1 plik) po oddaniu prowadzącemu zajęcia laboratoryjne musi zostać przesłany na adres `sdizo@zut.edu.pl` :

- plik z kodem źródłowym musi mieć nazwę: `nr_albumu.sdizo.lab06.main.c` (np. `123456.sdizo.lab06.main.c`); jeśli kod źródłowy programu składa się z wielu plików, to należy stworzyć jeden plik, umieszczając w nim kody wszystkich plików składowych; (plik może mieć rozszerzenie `.c` lub `.cpp`)
- plik musi zostać wysłany z poczty ZUT (`zut.edu.pl`);
- nagłówek maila (temat) musi mieć postać: `SDIZO IS1 XXXY LAB06`, gdzie `XXXY` to numer grupy (np. `SDIZO IS1 210C LAB06`);
- w pierwszych trzech liniach pliku z kodem źródłowym w komentarzach muszą znaleźć się:
  - linia 1: informacja identyczna z zamieszczoną w nagłówku maila
  - linia 2: imię i nazwisko
  - linia 3: adres email
- email **nie** powinien zawierać żadnej treści (tylko załącznik).