Zestaw 7

Słownik

Algorytmy i strukury danych I

> ***Słownik** przechowuje pary (klucz, wartość) i umożliwia dostęp do wartości za pomocą klucza.*

Szczegółowe informacje o tablicach z haszowaniem można znaleźć w [Cormen, 2013].

Zadanie 1. Słownik (`Dict.cpp`)

Zaimplementować słownik przy użyciu *tablicy z haszowaniem otwartym* wg poniższego schematu:

```
template<class K, class V>
class Dict {
using Pair = std::pair<K, V>;

    Dict();                          // Konstruktor
    clear();                         // Czyści słownik
    bool insert(const Pair& p);      // Dodaje parę klucz-wartość do słownika
    bool find(const K& k)       // Sprawdza czy słownik zawiera klucz
    V& operator[](const K& k); // Zwraca wartość dla klucza
    bool erase(const K& k);    // Usuwa parę od danym kluczu
    int size();                // Zwraca liczbę par
    bool empty();              // Sprawdza czy słownik jest pusty
    void buckets();            // Wypisuje informację o słowniku (patrz poniżej)
}
```

- Funkcja `insert` zwraca `true` jeżeli dodano nową parę lub `false` jeżeli para o danym kluczu już istnieje w słowniku
- Funkcja `find` zwraca `true` jeżeli para o kluczu `k` znajduje się w słowniku
- Operator `[]` zwraca referencję do nowej lub istniejącej wartości odpowiadającej kluczowi
- Funkcja `erase` zwraca `true` jeżeli usunięto parę o danym kluczu
- Funkcja `buckets` wypisuje na standardowe wyjście: znak '`#`', liczbę elementów w słowniku, liczbę klas, rozmiar najkrótszej i najdłuższej listy w klasach. Wszystkie wartości mają być wypisane w jednej linii zakończonej znakiem końca linii i oddzielone spacją.
- Należy napisać własną implementację funkcji `unsigned int hash(const K&)` dla konkretnego typu klucza, który jest wymagany. To ma być funkcja, a nie metoda klasy `Dict`.
- Należy użyć własnej implementacji listy wskaźnikowej i innych struktur, w miarę potrzeb zmodyfikować kod z poprzednich zestawów.

> ***Uwaga:** Haszowanie otwarte nie jest adresowaniem otwartym. W haszowaniu otwartym elementy przechowuje się w listach przyporządkowanych do klas, a w adresowaniu otwartym bezpośrednio w tablicy, ale niekoniecznie pod indeksem zwracanym przez funkcję mieszającą.*

Program `Dict.x` powinien wczytać do słownika `Dict<std::string, std::string>` pary słów z pliku o nazwie podanej jako argument linii komend (`argv[1]`).

Następnie wczytać słowa (klucze) ze standardowego wejścia. Jeżeli para o danym kluczu istnieje w słowniku należy wypisać odpowiadającą mu wartość, a w przeciwnym wypadku wypisać `-`.

Przykładowe pliki z danymi znajdują w folderze `Materiały/Zestaw07` na stronie ćwiczeń.

```
make
./Dict.x pairs.txt < input.txt > output.txt
diff -s result.txt output.txt
```

Zadanie 2. Słowa (slowa.txt)

W pliku slowa.txt umieścić 32 pary słów. Pierwsze słowo w parze ma być losowo wybranym, ale istniejącym, słowem w języku polskim i zaczynać się na tę sama literę jak Pana/Pani imię. Drugie słowo ma być angielskim tłumaczeniem pierwszego słowa (wybrać jedno znaczenie). W każdej linii znajduje się jedna para słów oddzielonych pojedynczą spacją. Słowa muszą być poprawne w danym języku, składać się tylko z małych litery alfabetu polskiego lub angielskiego, i nie mogą być nazwami własnymi. Plik zapisać w kodowaniu UTF-8.

Andrzej Görlich
a.goerlich@outlook.com