FEJLETT PROGRAMOZÁSI NYELVEK, 2009
11. GYAKORLAT – Hasító táblák

Cél:

- hasító függvények
- **`hash_set, hash_map, hash_multiset, hash_multimap`**

Készítsen egy-egy szótár osztályt a `set`, illetve a `hash_set` tárolók felhasználásával. A `hash_set` egy hasító táblával megvalósított tároló, amely megfelelően megválasztott hasító függvény segítségével gyors keresést tesz lehetővé. A `set` tárolóval ellentétben, ez nem teszi lehetővé a tárolt elemek rendezett sorrendjének előállítását. Ezt a tárolót nem tartalmazza a C++ szabvány, ennek ellenére mind a GNU C++ mind pedig a MS Visual C++ fordítók implementálják.

Az implementációt követően mérjük le mindkét osztály esetében a következőket:

- szótár felépítésének időigénye
- a szótárban található összes szó keresésének időigénye
- a szótárban található összes szó törlésének időigénye

A `hash_set` segítségével implementált szótár esetében mérje le az időt különböző hasító függvények használata esetében. Saját készítésű hasító függvény mellett érdemes kipróbálni a következőket:

`http://www.cse.yorku.ca/~oz/hash.html`

A szótár a szokásos: http://www.ms.sapientia.ro/~manyi/dict.txt

Adottak a következő fejállományok:

SetDictionary.h

```
#ifndef _SETDICTIONARY_H
#define _SETDICTIONARY_H

#include <string>
#include <set>
using namespace std;

class SetDictionary{
private:
    set<string> dict;
public:
    SetDictionary( string filename);
    bool addWord( string word );
    bool deleteWord( string word );
    bool findWord( string word );
    int size();
};

#endif /* _SETDICTIONARY_H */
```

FEJLETT PROGRAMOZÁSI NYELVEK, 2009
11. GYAKORLAT – Hasító táblák

HashSetDictionary.h

```
#ifndef _HASHSETDICTIONARY_H
#define _HASHSETDICTIONARY_H

#include <string>
#include <hash_set.h>//Ez kivetelesen igy mukodik

using namespace std;

//Opcionális sablonparaméter, implicit értéke: hash<T>, T: kulcs típusa
//Csak a standard C primitív típusaira nem kell implementálni
struct HashFunction{
    unsigned long operator()( const string& s ) const{
       //Implementálni!
    }

};

//Opcionális sablonparaméter, implicit értéke: equal_to<T>, T: kulcs típusa
struct EqualKey{
    bool operator()( const string& s1, const string& s2 )const{
       //Implementálni!

    }
};


class HashSetDictionary{
private:
    hash_set<string, HashFunction, EqualKey> dict;
public:
    HashSetDictionary( string filename);
    bool addWord( string word );
    bool deleteWord( string word );
    bool findWord( string word );
    int size();
};

#endif /* _HASHSETDICTIONARY_H */
```

Időmérés:

#include <ctime>

```
clock_t start, stop;
string word;

ifstream ifs("dict.txt");

//Felepites
start = clock();
HashSetDictionary dict2("dict.txt");
stop = clock();
cout<<"Dictionary construction: "<<(double)(stop-start)/ CLOCKS_PER_SEC<<endl;
```