

Проект 2

Задача 1.

Реализирайте свързано представяне на граф(за всеки връх се съхранява списък от наследниците му):

```
#include <iostream>
using namespace std;
template <typename T>
class graph{
public :
//включва връх в неявния граф
void addTop(const T&);
//изключва връх от неявния граф
void deleteTop(const T&);
//включва ребро от връх до връх в неявния граф
void addRib(const T&, const T&);
//изключва ребро от връх до връх в неявния граф
void deleteRib(const T&, const T&);
//проверява дали подадения параметър е връх в неявния граф
bool top(const T&);
//проверява дали има ребро от един връх до друг връх
bool rib(const T&, const T&);
//проверява дали неявният граф е празен
bool empty() const;
//намира указател към подаден връх в графа
elem_link1<T>* point(const T&);
//върща списък от върховете на неявния граф
LList<T> vertexes();
//извежда неявния граф
void print()const;
private:
LList < LList<T> > g;
};
```

Също така да се реализират функции, които обхождат графа в дълбочина(DFS) и в ширина(BFS), като имплементирате необходимите помощни функции, а също така и функция, която намира най-кратък път от един връх до друг.

Задача 2.

Реализирайте структурата от данни HashMap

```
const int HASH_TABLE_MAX_SIZE=1000;
template <typename KEY,VALUE>
struct HashEntry{
KEY key;
VALUE val;
};
template <typename KEY,VALUE>
class HashMap{
// в масив са запазени елементите,които се състоят от двойки ключ-стойност
HashEntry[HASH_TABLE_MAX_SIZE] table;
// дължина на таблицата
int nTable;
// hash key
unsigned int hashCode(KEY k);
public:
HashMap();
void insert(KEY, VALUE);
void erase(KEY);
~HashMap();
};
```

(Бонус) Задача 3.

Реализирайте алгоритъма бързо сортиране върху масив от целочислени стойности.