

Aula prática 10

Esta aula tem como objetivo estudar tabelas de dispersão (hash tables), e analisar a performance de algumas operações com as mesmas.

1. Pretende-se analisar o comportamento de tabelas de dispersão em comparação com listas ligadas. Para tal deverá analisar a biblioteca fornecida em **hash.cpp/hpp** bem como o ficheiro de teste **hash_test.cpp** e implementar as seguintes alíneas

- a) Implemente a função **hashFunctionkrm(string key)**. Esta função deverá calcular a hash com base na seguinte fórmula:

$$\text{hash}(i) = \text{hash}(i-1) + \text{key}[i], \text{ sendo } \text{hash}(0) = 7$$

A função deve receber uma string para a qual se pretende calcular a hash. Nota: $\text{key}[i]$ refere-se ao carácter na posição i da chave. A função deve devolver o resto da divisão da hash pelo tamanho da tabela.

- b) Implemente a função **search_value(string key, int func)** que deve pesquisar uma chave na tabela de dispersão. A função deve ter em atenção a função de hash utilizada para inserir na tabela bem como devolver o índice em que encontrou a chave ou -1 em caso de não ter encontrado.

- c) Implemente a função **deleteElement(string key, int func)** que apaga um elemento da tabela de dispersão. A função deve ter em atenção a função de hash utilizada para inserir na tabela.

- d) Complete o programa de teste de modo a que seja possível comparar a cronometração de ambas as funções de hash. Para cronometrar as funções pode usar a biblioteca `time` para obter a hora atual. Use o exemplo seguinte como referência:

```
clock_t beg, end;
double time;
beg = clock();
/* task */
end = clock();
tempo = (double)(fim - inicio) / CLOCKS_PER_SEC;
cout << "Time used by the first hash function is "<<
time << endl;
```

- e) Implemente a função **count_words_starting_char(char c)** que tem como objectivo contar o número de palavras começadas pelo carácter c, independentemente de este ser maiusculo ou minusculo e devolver essa mesma contagem.