

Trabalho 2

1 Descrição

Tabelas de Dispersão (do inglês *Hash Tables*) são estruturas de dados que associam chaves de pesquisa a valores com o intuito de garantir rapidez na busca de dados desejados. São tipicamente utilizadas quando o volume de elementos são grandes o suficiente para inviabilizar uma pesquisa como, por exemplo, em um dicionário ou em uma busca de páginas relevantes da *web*. Neste contexto, como ilustra a Figura 1, busca-se associar cada valor de entrada a um único índice da tabela por meio de uma função $f(x_{\text{input}}, k)$ (em que x_{input} é a entrada, ou chave, e k o número de posições disponíveis na tabela) chamada de função de espalhamento.

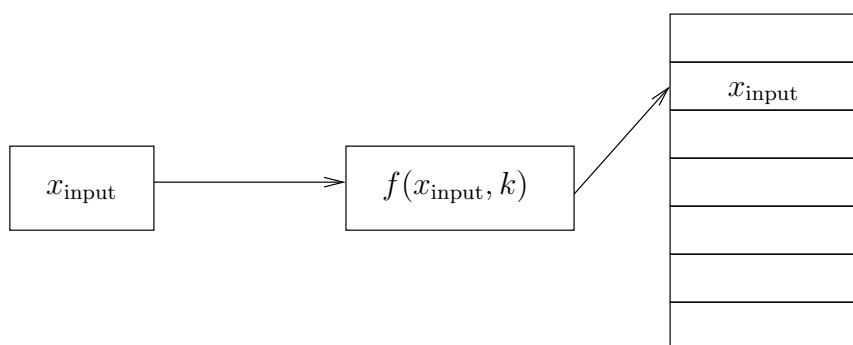


Figura 1: Referenciamento da Tabela de Espalhamento por meio da função de dispersão.

Neste trabalho uma Tabela de Dispersão deverá ser desenvolvida a partir de uma função de espalhamento comumente utilizada, definida pela operação de resto entre a chave e k , tal como apresentado na Equação 1, em que $h(x_{\text{input}}, k)$ é o índice da tabela que armazenará o valor x_{input} . Por exemplo, se é desejado armazenar o valor 73 em uma Tabela de Dispersão em que $k = 13$, calcula-se $73 \bmod 13 = 8$ e, portanto, armazena-se o número 73 na posição 8 da tabela. Note que o tamanho da Tabela de Dispersão será equivalente ao valor de k .

$$h(x_{\text{input}}, k) = x_{\text{input}} \bmod k \quad (1)$$

Contudo, apesar do objetivo das Tabelas de Dispersão ser o de produzir um índice único para cada valor de entrada, há a possibilidade da função de espalhamento apresentar índices iguais para diferentes dados, gerando o que se denomina como colisão. Uma das formas de contornar essa lacuna é, como ilustrado na Figura 2, construir um vetor v_p de ponteiros associados a vetores v_d que receberão os dados. Por exemplo, dada a chave $k = 13$ e duas entradas, 73 e 60 enviadas sequencialmente, v_d irá conter os dois valores na posição 8 de forma que $v_d[8] = \{73, 60\}$. Esses vetores v_d devem ser implementados de forma a terem tamanhos dinâmicos, ou seja, para cada valor a ser inserido em um vetor v_d , ele deve aumentar a sua capacidade de armazenamento.

Você deverá implementar uma Tabela de Dispersão com função de espalhamento definida pela Equação 1 de forma que as colisões sejam tratadas a partir do método citado anteriormente.

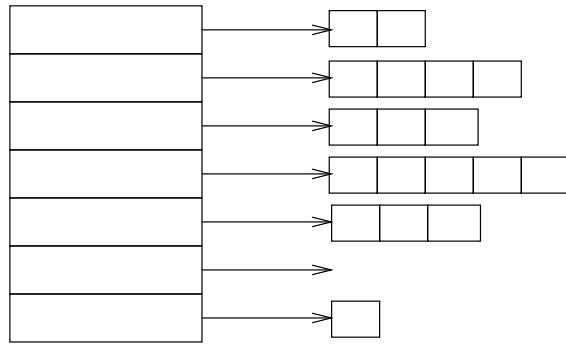


Figura 2: Estrutura utilizada para evitar colisões dos dados propagadas pela função de espalhamento.

Entrada	Saída
13	0:
9	1: 27 92
44	2:
45	3:
49	4: 95
70	5: 44 70
27	6: 45 97
73	7:
92	8: 73
97	9:
95	10: 49
	11:
	12:

Tabela 1: Tabela dos dados de entrada e saída do programa.

2 Informações Complementares

- Seu programa deverá receber como entrada o valor k e, logo após a quantidade de dados a serem armazenados e, posteriormente, os valores inteiros que serão armazenados;
- Seu programa deverá fornecer como saída a estrutura da tabela após os dados serem armazenados, tal como apresentado na Tabela 1. O primeiro valor das linhas deverá ser o índice da tabela e os restantes os valores associados;
- Os dados a serem armazenados não assumirão valores repetidos;
- A ordem dos elementos nos vetores v_d deve seguir a ordem de entrada dos elementos;
- Deve-se alocar dinamicamente as estruturas da tabela hash v_p e v_d (i.e., na memória *heap*). Utilize `realloc` para alocar v_d devido a sua variação de tamanho.