

Funções de dispersão perfeitas

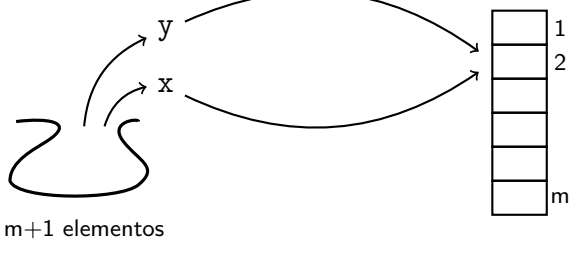
Nós acabamos de ver como é possível utilizar funções de dispersão (imperfeitas) para construir um filtro imperfeito.

Agora, nós vamos ver como é possível utilizar funções de dispersão imperfeitas para construir uma função de dispersão perfeita.

Quer dizer, uma função de dispersão perfeita é aquela onde nunca ocorre uma colisão.

Ora, mas isso é impossível.

Porque se a tabela tem tamanho m e nós temos um saco com $m + 1$ elementos, então certamente existem dois elementos no saco que são mapeados para a mesma posição da tabela



Certo.

Mas então a gente muda o problema.

Quer dizer, imagin que nós temos um saco com m elementos.

E imagine que a tabela tem tamanho m .

O problema é

→ Construir uma função de dispersão perfeita $H()$, que mapeia os m elementos do saco para m posições diferentes da tabela — (i.e., sem colisão)

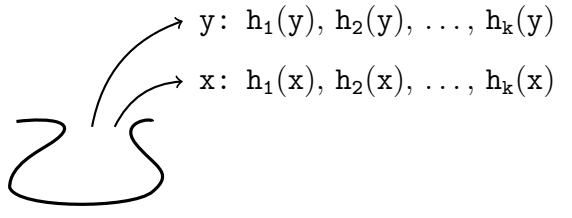
Bom, esse problema a gente consegue resolver.

E a ideia é resolvê-lo utilizando uma coleção de funções de dispersão imperfeitas.

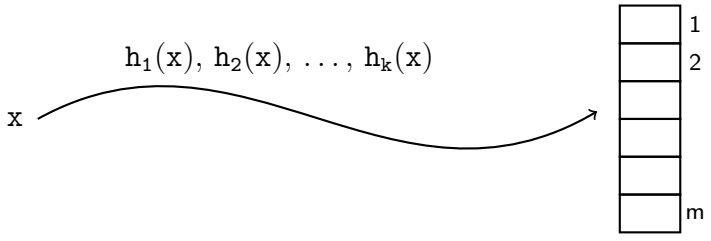
Quer dizer, as funções imperfeitas vão produzir colisões.

Mas elas vão produzir colisões de maneiras diferentes.

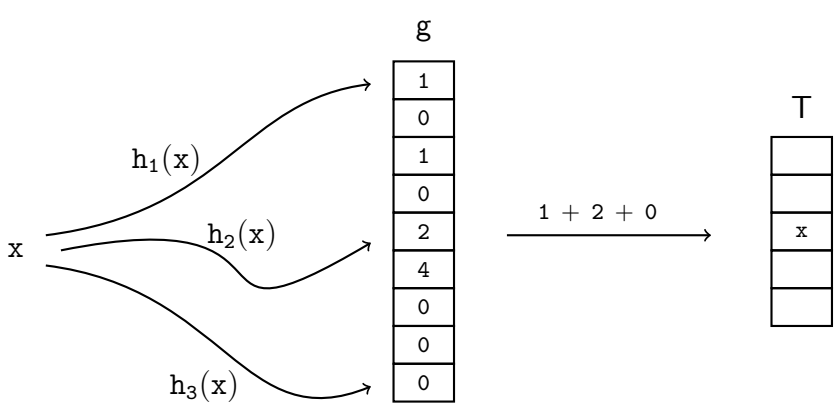
Daí que, ao aplicar as funções a todos os elementos do saco, cada um deles vai receber um conjunto de valores diferentes



E daí, a ideia é obter a posição de cada elemento na tabela T a partir da coleção de valores atribuídos pelas funções de dispersão.



Uma maneira simples de implementar essa ideia consiste em utilizar uma tabela intermediária g



Quer dizer, as funções h_1 , h_2 , h_3 indicam posições na tabela intermediária.

E a posição de x na tabela T é dada pela soma dos elementos de g que foram selecionados pelas funções h_1 , h_2 , h_3 .

Abaixo nós temos a definição das funções h_1 , h_2 , h_3 para 5 elementos x, y, z, w, u , que são mapeados todos para posições diferentes de T

	h_1	h_2	h_3	
x	1	5	9	$\rightarrow 1 + 2 + 0 = 3$
y	3	6	8	$\rightarrow 1 + 4 + 0 = 5$
w	1	4	8	$\rightarrow 1 + 0 + 0 = 1$
z	2	5	7	$\rightarrow 0 + 2 + 0 = 2$
u	2	6	8	$\rightarrow 0 + 4 + 0 = 4$

Quer dizer, as funções h_1 , h_2 , h_3 são imperfeitas e produzem colisões.

Mas a sua combinação por meio da tabela intermediária g resulta em um tabela de dispersão perfeita.

Legal, não é?

Bom, o truque é a escolha adequada dos valores que ficam armazenados na tabela intermediária g .

Quer dizer, tudo o que a gente espera das funções de dispersão $h_1, \dots, h_k$ é que elas tenha comportamento (aproximadamente) aleatório.

Daí, elementos diferentes vão selecionar coleções de posições diferentes na tabela g .

E daí, se não houverem coincidências numéricas, os elementos vão ser todos mapeados em posições diferentes da tabela T .

Sim.

Mas como é que a gente garante que não ocorrem coincidências numéricas?

Bom, é isso o que nós vamos ver a seguir.

Quer dizer, suponha que nós temos m elementos para serem mapeados nas m posições da tabela T .

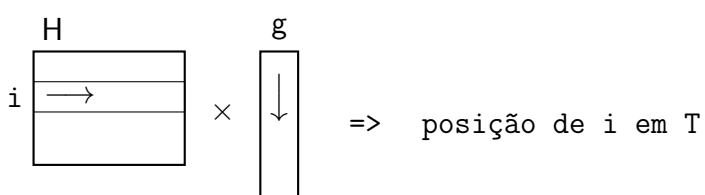
Suponha que nós vamos utilizar uma tabela intermediária g com tamanho N .

E suponha que as funções de dispersão $h_1, h_2, \dots, h_k$ já foram dadas.

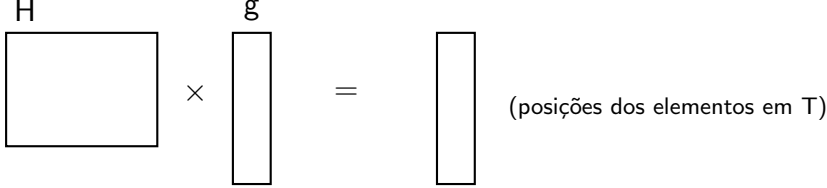
Daí, não é difícil construir a matriz

$$H(i,j) = \begin{cases} 1 & \text{se alguma função de dispersão seleciona a posição } j \text{ para o elemento } i \\ 0 & \text{caso contrário} \end{cases}$$

A ideia aqui é que a multiplicação da linha i da matriz H pelo vetor g vai produzir a posição do elemento i na tabela T

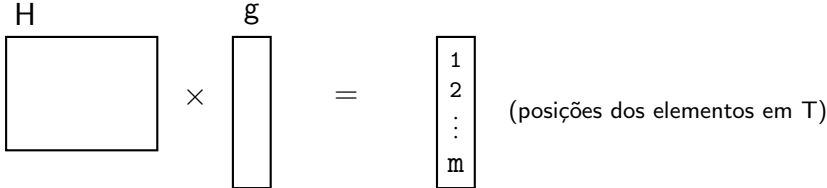


E a multiplicação da matriz inteira H pelo vetor g produz as posições de todos os elementos em T



A esperteza agora é escolher as posições em que a gente quer colocar os elementos na tabela T .

Por exemplo,



E daí, basta resolver esse sistema de equações lineares para determinar os valores de g que produzem essa distribuição dos elementos.

Se o vetor g é grande o suficiente, e as funções $h_1, \dots, h_k$ tem comportamento aleatório, o sistema tem solução com alta probabilidade

Não é legal?