

Espertezas quase perfeitas

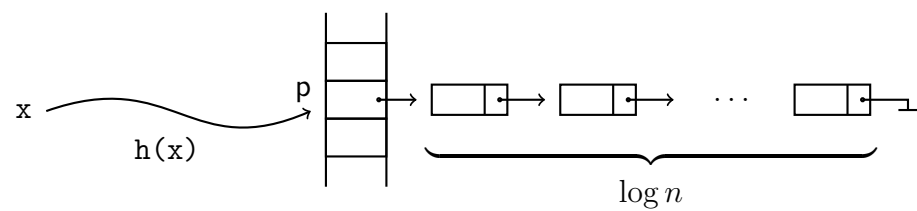
Certo.

Então o pior caso da busca é $O(\log n)$, tanto no esquema de endereçamento aberto como no esquema de encadeamento externo.

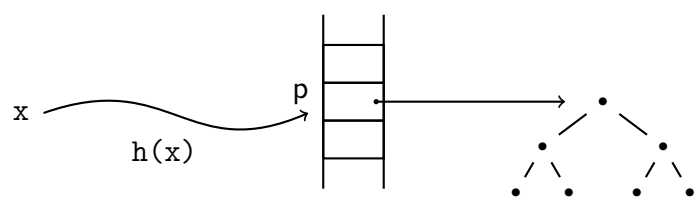
E a nossa tarefa agora é tentar melhorar isso.

Nós vamos concentrar a nossa atenção no esquema de encadeamento externo.

Nesse caso, o pior caso corresponde à maior lista encadeada externa



E aqui, nem é preciso pensar muito para imaginar que a lista encadeada pode ser substituída por uma árvore binária de busca (balanceada)



E apenas isso já é suficiente para reduzir o pior caso da busca para $O(\log \log n)$

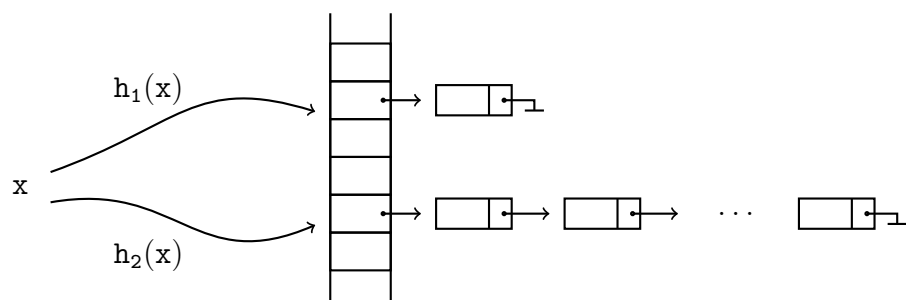
— (o que já é uma coisa bem pequenininha ...)

Mas, essa não é a única maneira de fazer as coisas.

Quer dizer, outra ideia consiste em utilizar duas funções de dispersão

$h_1(\cdot)$ $h_2(\cdot)$

Daí, no momento em que um novo elemento é inserido na tabela, nós utilizamos as duas funções



Quer dizer, a ideia aqui é que o novo elemento x é inserido na posição que tem a menor lista encadeada.

E a boa notícia é que esse esquema também reduz o pior caso da busca para $O(\log \log n)$.

Infelizmente, a análise desse esquema é complicada demais para apresentar aqui.

Mas, existe uma ideia ainda melhor do que essa.