

Funções de dispersão

Hmm ...

Para responder essa pergunta, nós fazemos outra pergunta

- *Como é que a gente acelera a busca binária?*

Bom, a gente já viu essa ideia nas primeiras aulas da disciplina.

Quer dizer, a ideia é você ter uma lista onde

- os números pares estão nas posições pares
 - os números ímpares estão nas posições ímpares
 - e tanto os pares como os ímpares estão em ordem crescente
- *(mas a lista inteira não precisa estar ordenada)*

Por exemplo,

	1	2	...												n
L	3	2	7	4	13	8	15	18	23	20	27	22			

Na prática, o que nós temos aqui são duas listas ordenadas com tamanho $n/2$.

Daí que, na prática nós fazemos a busca binária em uma lista de tamanho $n/2$.

E isso leva tempo

$\log_2(n/2) = \log_2 n - \log_2 2 = \log_2 n - 1$

Mas ainda tem um probleminha aqui.

Quer dizer, para saber em qual das listas a busca vai ser feita, é preciso verificar se o número é par ou ímpar.

E isso leva tempo $O(1)$.

Daí que o tempo total do processo é

$\log_2 n - 1 + O(1) = \log_2 n$

E nós acabamos não ganhando nada ...

E agora?

Bom, agora nós podemos tentar levar a nossa esperteza mais adiante.

Quer dizer, ao invés de trabalhar apenas com números pares e ímpares, nós podemos trabalhar com o resto da divisão por 4.

A ideia é que

- os números que deixam resto 1 são armazenados nas posições 1, 5, 9, 13, 17, ...
- os números que deixam resto 2 são armazenados nas posições 2, 6, 10, 14, 18, ...
- os números que deixam resto 3 são armazenados nas posições 3, 7, 11, 15, 19, ...
- os números que deixam resto 0 são armazenados nas posições 4, 8, 12, 16, 20, ...

E em cada um desses grupos os números estão organizados em ordem crescente.

Na prática, nós temos 4 listas ordenadas com tamanho $n/4$.

Daí que a busca binária leva tempo

$\log_2 n - 2$

E levando em conta o tempo $O(1)$ para calcular o resto da divisão por 4, isso nos dá

$\log_2 n - 2 + O(1) = \log_2 n - 1$

Legal, a coisa deu certo!

Mas não há qualquer razão para parar aqui.

Quer dizer, trabalhando com os restos da divisão por 8, o tempo da busca é reduzido para

$\log_2 n - 3$

E levando a coisa até o limite, nós podemos trabalhar com os restos da divisão por n .

Abaixo nós temos um exemplo com $n = 10$

	1	2	...							10
L	81	32	23	34	25	66	7	18	29	40

Quer dizer, à primeira vista a lista está toda bagunçada.

Mas basta olhar para o último dígito dos número para saber a sua posição na lista.

E isso reduz o tempo da busca para $O(1)$.

— *(é assim que você encontra as coisas no seu quarto?)*

Mas, não é difícil ver os problemas dessa solução.

Quer dizer, imagine que os 10 números que nós temos para armazenar são

13, 83, 53, 123, 133, 673, 543, 393, 703, 463

Ou então imagine que nós temos uma lista com apenas uma posição ocupada

	1	2	3	4	5	6	7	8	9	10
L							87			

E daí, alguém nos pede para armazenar o número 27.

Nesse caso, apesar da lista estar quase toda vazia, nós vamos dizer que não há espaço para ele.

E agora?

Bom, agora é a hora para uma pequena esperteza.

Quer dizer, e se ao invés de olhar para o último dígito do número, a gente olha para o penúltimo dígito?

Daí, os números acima seriam armazenados assim

	1	2	...							10
L	13	123	133	543	53	463	673	83	393	703

Ou então, a gente também pode somar todos os dígitos do número, e depois calcular o resto da divisão por 10.

Fazendo isso, nós obtemos o seguinte resultado

	1	2	...							10
L	83	543	463	13	393	673	133	53	123	703

Ou então, a gente pode pensar em alguma outra função qualquer.

Quer dizer, o truque é que a nossa função deve distribuir ou espalhar os números por todas as posições da lista.

Essas funções são conhecidas como *funções de dispersão* — *(ou hash functions, em inglês)*

Abaixo nós temos algumas funções de dispersão populares

- função quadrática: $h(x) = x^2 \bmod n$
- função multiplicativa:

Suponha que A é um número entre 0 e 1 — *(de preferência um número irracional)*

Daí o procedimento consiste em

- multiplicar o valor de x por A
- extrair a parte decimal do resultado
- multiplicar a parte decimal por n , e arredondar para baixo

Por exemplo, para $x = 1235$, $A = \sqrt{2}/2$, e $n = 15$

$\Rightarrow x \cdot A = 873,2768...$

$\Rightarrow 0,2768... \cdot 15 = 4$

E por aí vai ...