

UFPI - CCN - DC
Ciência da Computação
Estrutura de Dados

Tabela de Símbolos



Prof. Raimundo Moura
rsm@ufpi.edu.br

80

Tabela de Símbolos (TSs): Introdução

- ▶ Um *tabela de símbolos* é uma tabela com duas colunas: uma coluna de **chaves** (= keys) e uma de **valores** (= values). Dizemos que cada linha da tabela é um *item*. Cada item associa um valor a uma chave.

KEY	VALUE

81

Tabela de Símbolos (TSs): Exemplo

<i>chave</i>	<i>valor</i>
20219028326	ANA CAROLLINE SANTOS SILVA
20209065451	ANDRE LOPES ARAUJO
20219034771	EMILIANO DIAS SARMANHO DA
20219032543	ENZO BORGES ROCHA LIMA
20219004692	ENZO LEAL DE ARAGAO
20219033317	FELIPE CHAVES GOMES
...	...
20219032679	PAULO AUGUSTO MARTINS
20219036972	RODRIGO DA SILVA LEITE
20209053479	VICTOR MATHEUS BEZERRA LIMA
20219041275	VLADIMIR GADELHA DOS SANTOS

82

Tabela de Símbolos (TSs): Exemplo

<i>chave</i>	<i>valor</i>
www.ebay.com	66.135.192.87
www.princeton.edu	128.112.128.15
www.cs.princeton.edu	128.112.136.35
www.harvard.edu	128.103.60.24
www.yale.edu	130.132.51.8
www.cnn.com	64.236.16.20
www.google.com	216.239.41.99
www.nytimes.com	199.239.136.200
www.apple.com	17.112.152.32
www.slashdot.org	66.35.250.151
www.espn.com	199.181.135.201
www.weather.com	63.111.66.11
www.yahoo.com	216.109.118.65
...	...
www.ime.usp.br	143.107.45.37
www.ufpi.br	200.137.162.1

83

Tabela de Símbolos (TSs): Definição

- ▶ Uma *tabela de símbolos* (= *symbol table*) é um ADT que consiste em um conjunto de itens, sendo cada item um par (chave, valor), munido de duas operações fundamentais: *put*, que insere um novo item no conjunto, e *get*, que busca o valor associado a uma dada chave.

84

Tabela de Símbolos (TSs): Definição

- ▶ Nossas convenções sobre TSs:
 - não há chaves repetidas (as chaves são duas a duas distintas),
 - **null** nunca é usado como chave,
 - **null** nunca é usado como valor associado a uma chave.

85

Tabela de Símbolos: API (= Interface)

```
public class ST<Key,Value>
```

<code>ST()</code>	cria uma tabela de símbolos vazia
<code>void put(Key key, Value val)</code>	insere o item (key, val) nesta tabela
<code>Value get(Key key)</code>	retorna o valor associado a key
<code>boolean isEmpty()</code>	esta tabela está vazia?
<code>boolean contains (Key key)</code>	a chave key está nesta tabela?
<code>Iterable<Key> keys()</code>	lista todas as chaves desta tabela

OBS: Key e Value são tipos genéricos,
ou seja, parâmetros de tipo.

86

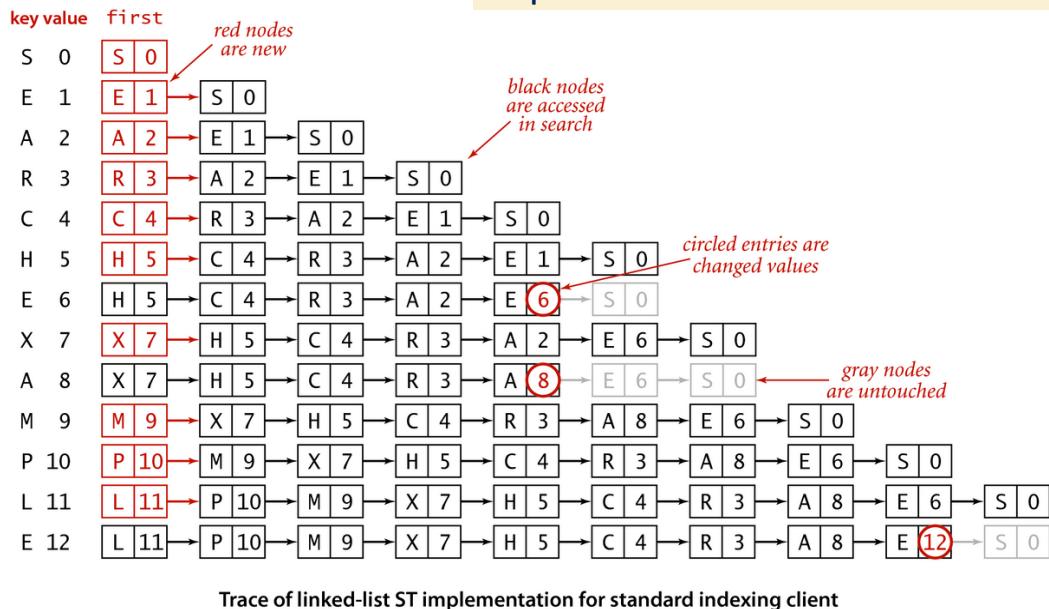
Tabela de Símbolos: API (= Interface)

- ▶ **Key** e **Value** são tipos genéricos, ou seja, parâmetros de tipo.
- ▶ Uma busca pode terminar em *acerto (search hit)* ou *falha (search miss)*. No primeiro caso, a busca é *bem-sucedida*; no segundo caso, a operação é *malsucedida* e `get` devolve `null`.
- ▶ Se a tabela já tem um item com chave igual a `key`, a operação `put(key, val)` substitui o item antigo pelo novo.

87

TS implementada com lista sequencial não ordenada

Input: S E A R C H E X A M P L E



88

Como estimar o desempenho de uma implementação de TS?

- ▶ Como estimar o tempo gasto em uma operação de inserção ou busca?
- ▶ O tempo é proporcional ao *número de chaves tocadas*.
- ▶ Diremos que o número de chaves tocadas durante uma operação é o ***custo*** da operação.

89

Como estimar o desempenho de uma implementação de TS?

- ▶ O *custo médio* de uma *busca bem-sucedida*, também conhecido como *custo de uma busca aleatória bem-sucedida*, é o quociente c/N , onde c é a soma dos custos das busca de todas as chaves na tabela e N é o número total de chaves na tabela.

90

Desempenho de SequentialSearchST

- ▶ Aplicar `SequentialSearchST` a uma lista com N chaves. Durante a execução de `get(k)` ou `put(k,v)`, uma chave da TS é *tocada* quando comparada com k . Quantas chaves o método `put()` toca? quantas `get()` toca?

	Nr. Máximo de Chaves Tocadas
<code>get()</code>	N
<code>put()</code>	N

91

Desempenho de SequentialSearchST

- ▶ Portanto, as duas operações, busca e inserção, são muito lentas. (A inserção é lenta em virtude de **nossa convenção** de não inserir chaves repetidas na TS)
- ▶ **Consequência:** Inserir N chaves distintas numa TS inicialmente vazia implementada em lista ligada consome $1 + 2 + \dots + N \cong N^2/2$ unidades de tempo no pior caso.

P.A. de razão 1: $1 + 2 + \dots + N$

$$\text{Logo, } S_{PA} = (1 + N)*N/2 = N/2 + N^2/2 \cong N^2/2$$

92

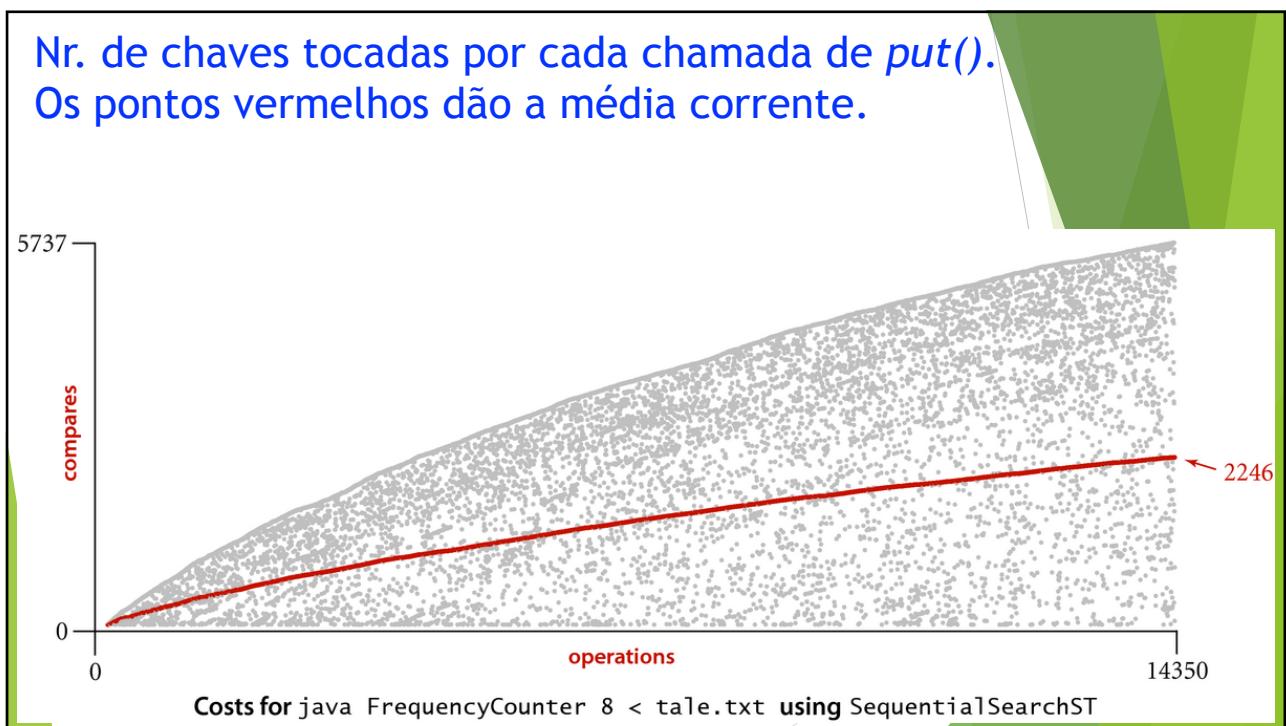
Desempenho de SequentialSearchST

- ▶ Como o **custo de uma busca aleatória bem-sucedida** é o **custoMédio/N**, temos:

$$\frac{N^2/2}{N} = \frac{N^2}{2} * \frac{1}{N} = \frac{N}{2}$$

93

Nr. de chaves tocadas por cada chamada de *put()*.
Os pontos vermelhos dão a média corrente.



94