

MAC0323 ALGORITMOS E ESTRUTURAS DE DADOS II

FOLHA DE SOLUÇÃO

Nome: Mateus Lotrova Stephanin

Número USP: 12542821

Assinatura

Mateus Lotrova Stephanin

Sua assinatura atesta a autenticidade e originalidade de seu trabalho e que você se compromete a seguir o código de ética da USP em suas atividades acadêmicas, incluindo esta atividade.

Exercício: E06 - Perguntas 1 e 2

Data: 06/06/2021

SOLUÇÃO

1) Como vimos numa aula do professor Yoshi, usando as funções de hashing presentes em SeparateChainingHashST, LinearProbingHashST e LinearProbingHashPrimesST, obtemos um mesmo valor de hashing(2112) para as strings "Aa" e "BB" e, também, para todas as strings que são combinações dessas duas (por exemplo: "AaBBaA" e "BBaAaBB"). Também vimos que, se vamos montar uma string com n blocos "Aa" ou "BB", há $\text{pow}(2, n)$ possíveis strings com mesmo valor de hashing. Assim, como são geradas 50000 (muito menor do que $\text{pow}(2, 20) = 1048576$) dessas strings na execução em questão, fica clara a razão da sua demora: toda vez que se insere uma string, deve-se percorrer todas as que já foram inseridas anteriormente (número altíssimo de displacements), visto que possuem o mesmo valor de hashing, o que nos dá um tempo linear em relação ao número de chaves inseridas (pior caso).

2) Para descobrir a fonte do problema, fiz a seguinte execução:

```
java-algs4 MGenerator 100000 20 ABCD BBBB 201905 |  
java-algs4 HashingProfile 7 -2 -u .
```

Olhando para os valores de $M(128)$, hits(8) e também, para a imagem gerada, fica claro que a imagem da função de hashing, nesse caso, é muito menor do que o contradomínio, o que nos leva a identificar a fonte do problema: similar à razão da pergunta 1, porém, em vez de um único valor de hashing reincidente, essa função de hashing mapeia as chaves apenas para alguns poucos valores, causando uma demora fora do comum.

Foi possível reproduzir esse problema com diversas outras entradas, entre elas:

```
1-java-algs4 MGenerator 50000 20 bola IIII 201905 | java-algs4  
HashingProfile 7 -2 -u .
```

Saída:

$M = 128$

$M \log M = 621.059873781711000$

$N = 48803$

$N/M = 381.2734375000000000$

hits = 8

```
2-java-algs4 MGenerator 100000 20 usp uuu 201905 |  
java-algs4 HashingProfile 7 -2 -u .
```

Saída:

$M = 128$

$M \log M = 621.059873781711000$

$N = 95304$

$N/M = 744.5625000000000000$

hits = 4