Universidade Federal do ABC
Algoritmos e Estruturas de Dados II 1^o Quadrimestre 2021

Prof. Carlos da Silva dos Santos

Projeto 01 (*individual*) – Funções de Espalhamento (*hashing*).

Data de entrega: até 23/03/2021, em atividade correspondente no Moodle.

1 Escopo do projeto

O primeiro projeto da disciplina envolve a implementação de funções de espalhamento. O projeto deve ser realizado individualmente e as linguagens de programação permitidas para a implementação do código são:

- C
- C++
- Java
- Javascript
- Python

Você deve implementar uma biblioteca (um módulo) que contenha as seguintes funções:

- Função de espalhamento pelo método da divisão
- Função de espalhamento pelo método da multiplicação

Além da implementação das funções acima, você deve realizar alguns experimentos e entregar um relatório, conforme descrito nas próximas seções. O principal mecanismo de avaliação do projeto será o relatório. Leia atentamente as instruções a seguir para formatar corretamento o relatório e demais artefatos que você deve entregar.

2 Especificação do código

As funções principais do projeto (funções de espalhamento) devem ser reunidas em um módulo que permita a reutilização do código. O módulo e cada função individual devem ser documentados.

2.1 Descrição das funções

Função de espalhamento pelo método da divisão: no método da divisão, a função de espalhamento é definida como

$$h(k) = k \mod m \tag{1}$$

em que k é um número natural (chave), $h(k)$ é a função de espalhamento e m é um número natural. Os valores de $h(k)$ ficam então restritos ao intervalo $\{0, 1, 2, \ldots, (m-1)\}$. Em geral, recomenda-se que m seja escolhido como um número primo, não muito próximo de alguma potência de 2.

Você deve implementar uma função para calcular $h(k)$ seguindo a fórmula 1, a função deve receber um número inteiro `key` (chave) e um inteiro `m` (parâmetro da fórmula 1). Para implementar a função, reflita sobre quais valores de `key` e `m` devem ser considerados válidos (p. ex. o que acontece quando `m` é negativo?). Tente usar os mecanismos adequados da linguagem escolhida (códigos de erro, exceções, etc) para tratar combinações de parâmetros que levem a resultados inválidos/indefinidos.
Você deve realizar uma série de experimentos para testar o funcionamento da sua função:

(a) Usando $m = 12$, faça um programa que teste sua função com valores de chave variando de 0 até 100. Quando o resultado $h(x)$ for igual a 3, imprima o valor de chave correspondente. Você consegue notar um padrão para esses valores de chave?

(b) Repita o item anterior com $m = 11$ e imprimindo as chaves que resultam em $h(x) = 3$. Você consegue notar um padrão para esses valores de chave?

(c) Usando $m = 97$ (um número primo) conte o número de colisões para cada valor diferente de $h(k)$, usando chaves no intervalo $\{1, 2, 3, \ldots, 10000\}$. *Dica*: você pode acumular as contagens em um vetor de m posições, inicialmente preenchido com zeros. A cada vez que você calcular um novo valor $h(k)$, incremente a posição correspondente no vetor de contagens. Salve os resultados dessas contagens em um arquivo e faça um gráfico de *número de colisões* em função do valor do *hash*. Você pode salvar as contagens em um arquivo de *valores separados por vírgula*, em que cada linha tem o formato `chave, contagem`. O gráfico pode ser construído em um programa qualquer de planilhas, por exemplo.

Função de espalhamento pelo método da multiplicação: no método da multiplicação, a função de espalhamento é definida como

$$h(k) = \lfloor m \times ((k \times A) \mod 1) \rfloor \tag{2}$$

Em que k é a chave, m é o número de posições da tabela, A é uma constante não negativa, $0 < A < 1$. O símbolo $\lfloor x \rfloor$ representa a função *chão*, isto é, o maior inteiro que seja menor que x (arredondamento de x "para baixo").
Você deve implementar uma função para calcular $h(k)$, de acordo com a fórmula 2. A função deve receber um inteiro `key` (chave), um inteiro `m` e um valor em ponto flutuante `a`, correspondendo aos parâmetros da fórmula acima.
Para implementar a função, reflita sobre quais valores de `key`, `m` e `a` devem ser considerados válidos. Tente usar os mecanismos adequados da linguagem escolhida (códigos de erro, exceções, etc) para tratar combinações de parâmetros que levem a resultados inválidos/indefinidos.

(a) Usando $m = 200$ e $A = 0.62$, faça um programa que teste sua função com valores de chave variando de 1 até 500 mil. Conte o número de colisões para cada valor diferente de $h(k)$. Salve os resultados dessas contagens em um arquivo e faça um gráfico de *número de colisões* em função do valor do *hash*.

(b) Usando $m = 200$ e $A = 0.61803398875$ (número derivado da razão áurea), repita o item anterior. Compare os resultados de distribuição das colisões.

2.2 Entrega do código

Você pode entregar o código como um arquivo *.zip* contendo toda a árvore de diretórios (pastas) do seu código. **Por favor, use o formato .zip, não use outros tipos de compactação**.

Alternativamente, você pode manter seu código em algum repositório público (github, gitlab, etc) e entregar a URL do seu repositório (nesse caso, será considerada para avaliação a versão do código mais recente na data de entrega, alterações posteriores à data de entrega não serão consideradas). Se optar por manter o código em um repositório público, identifique a URL de maneira clara no seu relatório.
Entregue apenas os arquivos fonte do seu código, não inclua arquivos compilados. O seu código deve ser acompanhado de um arquivo texto, contendo instruções de uso e eventuais instruções de compilação. Este arquivo texto deve identificar a estrutura do seu código, documentando a localização do módulo contendo as funções e explicando como executar os programas que demonstram o funcionamento das funções de espalhamento.

3 Relatório

Você deve redigir e entregar um relatório sucinto, contendo os gráficos de números de colisões e reportando os resultados dos experimentos descritos nas seções anteriores.

4 Critérios de avaliação do projeto

O projeto será avaliado em uma escala binária: *aceito*/*não aceito*. Para que o projeto seja aceito, todos os itens seguintes devem ser observados:

- É necessário entregar tanto o código das funções e programas quanto o relatório.
- O código fonte de todas as funções e dos programas descritos na Seção 2.1 devem ser entregues.
- As funções de espalhamento devem ser implementadas em um módulo (biblioteca) independente, que permita importar/carregar/*linkar* as funções em algum outro programa.
- O relatório deve conter todos os gráficos de quantidade de colisões descritos na Seção 2.1.
- O relatório deve conter respostas sucintas para as questões formuladas na Seção 2.1, bem como comentários sobre o resultado de cada experimento descrito na mesma seção.

Quaisquer dúvidas sobre o enunciado em geral e sobre os critérios de avaliação devem ser resolvidas até 2 dias antes da data de entrega, em consulta ao docente.