

## Atividade introdutória 01 e 02

### Algoritmos como uma tecnologia

Suponha que os computadores fossem infinitamente rápidos e que a memória do computador fosse livre.

- 1) **Você teria alguma razão para estudar algoritmos? Discuta com os seus colegas do grupo e depois escreva uma reflexão sobre a situação.**

#### Resposta/reflexão:

Se os computadores fossem infinitamente rápidos, qualquer método correto para revolver um problema serviria. É provável que você quisesse que sua implementação estivesse dentro dos limites da boa engenharia de software (i.e., que ela fosse bem documentada e projetada) mas, com maior frequência, você utilizaria o método que fosse o mais fácil de implementar.

É claro que os computadores podem ser rápidos, mas não são infinitamente rápidos. A memória pode ser de baixo custo, mas não é gratuita. Assim, o tempo de computação é um recurso limitado, bem como o espaço na memória. Esses recursos devem ser usados de forma sensata, e algoritmos eficientes em termos de tempo ou espaço ajudarão você a usá-los.

#### Eficiência

Algoritmos criados para resolver o mesmo problema muitas vezes diferem de forma drástica em sua eficiência. Essas diferenças podem ser muito mais significativas que as diferenças relativas a hardware e software.

Tomando como exemplo dois algoritmos para ordenação. O primeiro, conhecido como ordenação por inserção, leva um tempo aproximadamente igual a  $c_1 n^2$  para ordenar  $n$  itens, onde  $c_1$  é uma constante que não depende de  $n$ . Isto é, ela demora um tempo aproximadamente proporcional a  $n^2$ . O segundo, de ordenação por intercalação, leva um tempo aproximadamente igual a  $c_2 n \lg n$ , onde  $\lg n$  representa  $\log_2 n$  e  $c_2$  é outra constante que também não depende de  $n$ . A ordenação por inserção normalmente tem um fator constante menor que a ordenação por intercalação, e assim,  $c_1 < c_2$ . Todavia, tais constantes **podem** ser muito menos significativos no tempo de execução que a dependência do tamanho da entrada  $n$ . Onde a ordenação por intercalação tem um fator  $\lg n$  em seu tempo de execução, a ordenação por inserção tem um fator  $n$ , que é muito maior. Embora a ordenação por inserção em geral seja mais rápida que a ordenação por intercalação para pequenos tamanhos de entradas, uma vez que o tamanho da entrada  $n$  se tornar grande o suficiente, a vantagem da ordenação por intercalação de  $\lg n$  contra  $n$  compensará com sobras a diferença em fatores constantes. Independente do quanto  $c_1$  seja menor que  $c_2$ , sempre haverá um ponto de passagem além do qual a ordenação por intercalação será mais rápida.

Como um exemplo concreto, vamos comparar um computador mais rápido (computador A) que executa a ordenação por inserção com um computador mais lento (computador B) que executa a ordenação por intercalação. Cada um deles deve ordenar um arranjo de um milhão de números.

Suponha que o computador A execute um bilhão de instruções por segundo e o computador B execute apenas dez milhões de instruções por segundos, assim, o computador A será 100 vezes mais rápido que o computador B em capacidade bruta de computação. Para tornar a diferença ainda mais drástica, suponha que o programador mais astucioso do mundo codifica a ordenação por inserção em linguagem de máquina para o computador A, e que o código resultante exija  $2n^2$  instruções para ordenar  $n$  números. (Aqui,  $c_1 = 2$ ). Por outro lado, a ordenação por intercalação é programada para o computador B por um programador médio que utiliza uma linguagem de alto nível com um compilador ineficiente, com o código resultante totalizando  $50n \lg n$  instruções (de forma que  $c_2 = 50$ ). Para ordenar um milhão de números, o computador A demora

$$2 \times (10^6)^2 \text{ instruções} / 10^9 \text{ instruções/segundo} = 2000 \text{ segundos}$$

enquanto o computador B demora

$$50 \times 10^6 \lg 10^6 \text{ instruções} / 10^7 \text{ instruções/segundo} \approx 100 \text{ segundos}$$

- 2) **Dado o caso concreto descrito acima, discuta-o com os seus colegas do grupo e depois escrevam uma reflexão sobre a situação.**

**Resposta/reflexão:**

- 3) **Vamos explorar um ponto mais dessa situação concreta: como desenrolaria essa situação da ordenação por intercalação quando você ordenasse um arranjo de dez milhões de números:**

- a) calcule o tempo de ordenação por inserção.
- b) calcule o tempo de ordenação por intercalação.
- c) compare o resultado e comente o resultado da comparação.

Para responder os itens (a) e (b), escreva a fórmula do cálculo e sua aplicação com os seus valores. **Discuta os resultados com os seus colegas do grupo e depois escrevam uma reflexão sobre a situação.**

**Resposta item 3a):**

**Resposta item 3b):**

**Resposta item 3c):**

O exemplo anterior mostra que os algoritmos, como o hardware de computadores, constituem uma **tecnologia**. O desempenho total do sistema depende da escolha de algoritmos eficientes tanto quanto da escolha de hardware rápido. Da mesma maneira que estão havendo rápidos avanços em outras tecnologias computacionais, eles também estão sendo obtidos em algoritmos.

**4) Os algoritmos são verdadeiramente tão importantes nos computadores contemporâneos em comparação com outras tecnologias avançadas, como:**

- Hardware com altas taxas de *clock*, pipelines e arquiteturas *superescalares*.
- Interfaces gráficas do usuário (GUIs) intuitivas e fáceis de usar.
- Redes locais e remotas.
- Sistemas orientados a objetos.
- Computação em nuvem.

**Discuta as tecnologias acima com os seus colegas do grupo e depois escrevam uma reflexão sobre a situação, em particular, a última tecnologia da lista acima.**

**Resposta/reflexão:**