UNIVERSIDADE FEDERAL DE SERGIPE

DEPARTAMENTO DE COMPUTAÇÃO

Força bruta e indução matemática

Projeto e Análise de Algoritmos

Bruno Prado

Departamento de Computação / UFS

Introdução

- O que é força bruta?
  - É um método direto para resolução de um problema

Introdução

- O que é força bruta?
  - É um método direto para resolução de um problema
  - Geralmente utiliza as técnicas mais simples

Introdução

- O que é força bruta?
  - É um método direto para resolução de um problema
  - Geralmente utiliza as técnicas mais simples
  - Baseado na própria definição do problema

Introdução

- O que é indução matemática?
  - É um método de prova matemática

Introdução

- O que é indução matemática?
  - É um método de prova matemática
  - Estabelece que uma proposição é verdadeira para o conjunto dos números naturais

Introdução

- O que é indução matemática?
  - É um método de prova matemática
  - Estabelece que uma proposição é verdadeira para o conjunto dos números naturais
  - Intuição: uma sequência de dominós caindo

Força bruta

- Problema de avaliação da expressão a^n
  - Definição: $f(a,n) = \overbrace{a \times \ldots \times a}^{n}$
  - Com a constante e $n \in \mathbb{N}$

```c
// Padrão de tipos por tamanho
#include <stdint.h>
// Exponenciação por força bruta
int64_t exp_fb(int32_t a, uint32_t n) {
    // r = 1
    int64_t r = 1;
    // n iterações
    for(uint32_t i = 0; i < n; i++)
        // a * ... * a
        r = r * a;
    // r = a^n
    return r;
}
```

Força bruta

- Problema de avaliação da expressão a^n
  - Definição: $f(a, n) = \overbrace{a \times \ldots \times a}^{n}$
  - Com a constante e $n \in \mathbb{N}$

```c
// Padrão de tipos por tamanho
#include <stdint.h>
// Exponenciação por força bruta
int64_t exp_fb(int32_t a, uint32_t n) {
    // r = 1
    int64_t r = 1;
    // n iterações
    for(uint32_t i = 0; i < n; i++)
        // a * ... * a
        r = r * a;
    // r = a^n
    return r;
}
```

Espaço $\Theta(1)$ e tempo $\Theta(n)$

Força bruta

- Problema de avaliação da expressão a^n
  - Aplicando a estratégia de exponenciação binária
  - Definição: $f(a,n) = \begin{cases} a & n = 1 \\ a \times f(a^2, \frac{n-1}{2}) & n \; impar \; , n > 0 \\ f(a^2, \frac{n}{2}) & n \; par \end{cases}$
  - Com a constante e $n \in \mathbb{N}$

Força bruta

- Problema de avaliação da expressão a^n
  - Aplicando a estratégia de exponenciação binária

```c
// Padrão de tipos por tamanho
#include <stdint.h>
// Exponenciação binária recursiva
int64_t exp_bin_r(int32_t a, uint32_t n) {
    // Caso base
    if(n == 1) return a;
    // n é impar
    else if(n % 2) return a * exp_bin_r(a * a, (n -
        1) / 2);
    // n é par
    else return exp_bin_r(a * a, n / 2);
}
```

Força bruta

- Problema de avaliação da expressão a^n
  - Aplicando a estratégia de exponenciação binária

```c
// Padrão de tipos por tamanho
#include <stdint.h>
// Exponenciação binária recursiva
int64_t exp_bin_r(int32_t a, uint32_t n) {
    // Caso base
    if(n == 1) return a;
    // n é impar
    else if(n % 2) return a * exp_bin_r(a * a, (n -
        1) / 2);
    // n é par
    else return exp_bin_r(a * a, n / 2);
}
```

Espaço e tempo $O(\log_2 n)$

Força bruta

- Problema de avaliação da expressão a^n
  - Aplicando a estratégia de exponenciação binária

```c
// Padrão de tipos por tamanho
#include <stdint.h>
// Exponenciação binária iterativa
int64_t exp_bin_i(int32_t a, uint32_t n) {
    // r = 1
    int64_t r = 1;
    // Iterando enquanto n > 0
    while(n > 0) {
        // Ajuste se n for ímpar
        if(n % 2) { r = r * a; n = n - 1; }
        // a^2m = a^m * a^m
        a = a * a; n = n / 2;
    }
    // r = a^n
    return r;
}
```

Força bruta

- Problema de avaliação da expressão a^n
  - Aplicando a estratégia de exponenciação binária

```c
// Padrão de tipos por tamanho
#include <stdint.h>
// Exponenciação binária iterativa
int64_t exp_bin_i(int32_t a, uint32_t n) {
    // r = 1
    int64_t r = 1;
    // Iterando enquanto n > 0
    while(n > 0) {
        // Ajuste se n for ímpar
        if(n % 2) { r = r * a; n = n - 1; }
        // a^2m = a^m * a^m
        a = a * a; n = n / 2;
    }
    // r = a^n
    return r;
}
```

Espaço $\Theta(1)$ e tempo $O(\log_2 n)$

Força bruta

- Por que e quando usar força bruta?
  - Soluções fáceis e simples de implementar

Força bruta

- Por que e quando usar força bruta?
  - Soluções fáceis e simples de implementar
  - Conjunto de entrada de tamanho muito pequeno

Força bruta

- Por que e quando usar força bruta?
  - Soluções fáceis e simples de implementar
  - Conjunto de entrada de tamanho muito pequeno
  - Propósito educacional e entendimento do problema

Indução

- Estruturação do método de prova
  - Caso base
    - Utiliza valores iniciais, geralmente os menores possíveis
    - Para $n \geq 0$, o caso base é $P(0)$
    - Demonstra seu funcionamento básico

Indução

- Estruturação do método de prova
  - Caso base
    - Utiliza valores iniciais, geralmente os menores possíveis
    - Para $n \geq 0$, o caso base é $P(0)$
    - Demonstra seu funcionamento básico

$$P(n) = 0 + 1 + \ldots + n \stackrel{?}{=} \frac{n \times (n+1)}{2}$$

Indução

- Estruturação do método de prova
  - Caso base
    - Utiliza valores iniciais, geralmente os menores possíveis
    - Para $n \geq 0$, o caso base é $P(0)$
    - Demonstra seu funcionamento básico

$$P(n) = 0 + 1 + \ldots + n \stackrel{?}{=} \frac{n \times (n+1)}{2}$$

$$P(0) = 0 \quad \stackrel{?}{=} \quad P(0) = \frac{0 \times (0+1)}{2}$$

Indução

- Estruturação do método de prova
  - Caso base
    - Utiliza valores iniciais, geralmente os menores possíveis
    - Para $n \geq 0$, o caso base é $P(0)$
    - Demonstra seu funcionamento básico

$$P(n) = 0 + 1 + \ldots + n \stackrel{?}{=} \frac{n \times (n+1)}{2}$$

$$P(0) = 0 \quad \stackrel{?}{=} \quad P(0) = \frac{0 \times (0+1)}{2}$$

$$0 \quad \stackrel{V}{=} \quad 0$$

Indução

- Estruturação do método de prova
  - Passo indutivo
    - Utiliza-se um valor $k > 0$ arbitrário como hipótese

$$P(n) = 0 + 1 + \ldots + n \stackrel{?}{=} \frac{n \times (n+1)}{2}$$

Indução

- Estruturação do método de prova
  - Passo indutivo
    - Utiliza-se um valor $k > 0$ arbitrário como hipótese

$$P(n) = 0 + 1 + \ldots + n \stackrel{?}{=} \frac{n \times (n+1)}{2}$$

$$P(k) = 0 + 1 + \ldots + k \quad \stackrel{?}{=} \quad P(k) = \frac{k \times (k+1)}{2}$$

Indução

- Estruturação do método de prova
  - Passo indutivo
    - Utiliza-se um valor $k > 0$ arbitrário como hipótese

$$P(n) = 0 + 1 + \ldots + n \stackrel{?}{=} \frac{n \times (n+1)}{2}$$

$$P(k) = 0 + 1 + \ldots + k \quad \stackrel{?}{=} \quad P(k) = \frac{k \times (k+1)}{2}$$

$$0 + 1 + \ldots + k \quad \stackrel{?}{=} \quad \frac{k \times (k+1)}{2}$$

Indução

- Estruturação do método de prova
  - Passo indutivo
    - Assumimos a hipótese como verdadeira
    - Aplica-se o valor $k+1$ confirmar a tese

$$P(k+1) = 0 + 1 + \ldots + k + (k+1) \quad \stackrel{?}{=} \quad P(k+1)$$

Indução

- Estruturação do método de prova
  - Passo indutivo
    - Assumimos a hipótese como verdadeira
    - Aplica-se o valor $k+1$ confirmar a tese

$$
\begin{aligned}
P(k+1) = 0 + 1 + \ldots + k + (k+1) &\overset{?}{=} P(k+1) \\
P(k) + (k+1) &\overset{?}{=} \frac{(k+1)^2 + (k+1)}{2}
\end{aligned}
$$

Indução

- Estruturação do método de prova
  - Passo indutivo
    - Assumimos a hipótese como verdadeira
    - Aplica-se o valor $k+1$ confirmar a tese

$$
\begin{aligned}
P(k+1) = 0 + 1 + \ldots + k + (k+1) &\stackrel{?}{=} P(k+1) \\
P(k) + (k+1) &\stackrel{?}{=} \frac{(k+1)^2 + (k+1)}{2} \\
\frac{k \times (k+1)}{2} + (k+1) &\stackrel{?}{=} \frac{(k+1)^2 + (k+1)}{2}
\end{aligned}
$$

Indução

- Estruturação do método de prova
  - Passo indutivo
    - Assumimos a hipótese como verdadeira
    - Aplica-se o valor $k + 1$ confirmar a tese

$$\begin{aligned}
P(k+1) = 0 + 1 + \ldots + k + (k+1) &\stackrel{?}{=} P(k+1) \\
P(k) + (k+1) &\stackrel{?}{=} \frac{(k+1)^2 + (k+1)}{2} \\
\frac{k \times (k+1)}{2} + (k+1) &\stackrel{?}{=} \frac{(k+1)^2 + (k+1)}{2} \\
\frac{k^2 + k + k + 1 + (k+1)}{2} &\stackrel{?}{=} \frac{(k+1)^2 + (k+1)}{2}
\end{aligned}$$

Indução

- Estruturação do método de prova
  - Passo indutivo
    - Assumimos a hipótese como verdadeira
    - Aplica-se o valor $k + 1$ confirmar a tese

$$
\begin{aligned}
P(k+1) = 0 + 1 + \ldots + k + (k+1) &\stackrel{?}{=} P(k+1) \\
P(k) + (k+1) &\stackrel{?}{=} \frac{(k+1)^2 + (k+1)}{2} \\
\frac{k \times (k+1)}{2} + (k+1) &\stackrel{?}{=} \frac{(k+1)^2 + (k+1)}{2} \\
\frac{k^2 + k + k + 1 + (k+1)}{2} &\stackrel{?}{=} \frac{(k+1)^2 + (k+1)}{2} \\
\frac{(k+1)^2 + (k+1)}{2} &\stackrel{V}{=} \frac{(k+1)^2 + (k+1)}{2}
\end{aligned}
$$

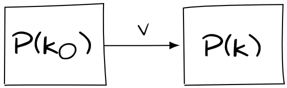
Indução

- Estruturação do método de prova
  - Caso base: $P(k_0)$
  - Hipótese: $P(k)$
  - Tese: $P(k+1)$

$$\boxed{P(k_0)}$$

Indução

- Estruturação do método de prova
  - Caso base: $P(k_0)$
  - Hipótese: $P(k)$
  - Tese: $P(k+1)$



Indução

- Estruturação do método de prova
  - Caso base: $P(k_0)$
  - Hipótese: $P(k)$
  - Tese: $P(k+1)$

Indução

- Estruturação do método de prova
  - Caso base: $P(k_0)$
  - Hipótese: $P(k)$
  - Tese: $P(k+1)$

$$\boxed{P(k_0)} \xrightarrow{\vee} \boxed{P(k)} \xrightarrow{\vee} \boxed{P(k+1)}$$

$\forall n \geq k_0$ a proposição $P(k)$ é verdadeira

Indução

- Definições de prova por indução
  - Fraca
    - Somente o passo k é considerado na hipótese
    - $P(k_0) \land \forall k \geq k_0\, (P(k) \longrightarrow P(k+1))$
    - É preciso demonstrar que o caso base $P(k_0)$ e a hipótese indutiva $P(k) \longrightarrow P(k+1)$ são verdadeiras

Indução

- Definições de prova por indução
  - Fraca
    - Somente o passo k é considerado na hipótese
    - $P(k_0) \wedge \forall k \geq k_0 \, (P(k) \longrightarrow P(k+1))$
    - É preciso demonstrar que o caso base $P(k_0)$ e a hipótese indutiva $P(k) \longrightarrow P(k+1)$ são verdadeiras
  - Forte
    - Os passos anteriores até k são utilizados na hipótese
    - $P(k_0) \wedge P(k_1) \wedge \ldots \wedge P(k) \longrightarrow P(k+1)$
    - Além do caso base, todas as proposições menores ou iguais à k são verdadeiras

Indução

- Definições de prova por indução
  - Fraca
    - Somente o passo k é considerado na hipótese
    - $P(k_0) \land \forall k \geq k_0\,(P(k) \longrightarrow P(k+1))$
    - É preciso demonstrar que o caso base $P(k_0)$ e a hipótese indutiva $P(k) \longrightarrow P(k+1)$ são verdadeiras
  - Forte
    - Os passos anteriores até k são utilizados na hipótese
    - $P(k_0) \land P(k_1) \land \ldots \land P(k) \longrightarrow P(k+1)$
    - Além do caso base, todas as proposições menores ou iguais à k são verdadeiras

Ambas as definições são logicamente equivalentes

Exemplo

- Prove que $\forall n \geq 0 \rightarrow 3^n > n^2$ é verdadeira
  - Caso base
    - $n = 0$ temos $3^0 > 0^2 \rightarrow 1 > 0$ é verdadeira
    - $n = 1$ temos $3^1 > 1^2 \rightarrow 3 > 1$ é verdadeira

Exemplo

- Prove que $\forall n \geq 0 \rightarrow 3^n > n^2$ é verdadeira
  - Caso base
    - $n = 0$ temos $3^0 > 0^2 \rightarrow 1 > 0$ é verdadeira
    - $n = 1$ temos $3^1 > 1^2 \rightarrow 3 > 1$ é verdadeira
  - Passo indutivo
    - Para um valor de $k > 1$ é obtida a hipótese $3^k > k^2$
    - Assume-se que a hipótese é verdadeira
    - Demonstrar que a tese $3^{k+1} > (k+1)^2$ é verdadeira

Exemplo

- Prove que $\forall n \geq 0 \rightarrow 3^n > n^2$ é verdadeira
  - Hipótese: $3^k > k^2$
  - Tese: $3^{k+1} > (k+1)^2$
  - Multiplicando ambos os lados da hipótese por 3

$$\begin{aligned} 3^k &> k^2 \\ 3 \times 3^k &> 3k^2 \\ 3^{k+1} &> 3k^2 \end{aligned}$$

Exemplo

- Prove que $\forall n \geq 0 \rightarrow 3^n > n^2$ é verdadeira
  - Se $k > 1$, temos que $3k^2 = k^2 + k^2 + k^2 > k^2 + 2k + 1 = (k+1)^2$

Exemplo

- Prove que $\forall n \geq 0 \rightarrow 3^n > n^2$ é verdadeira
  - Se $k > 1$, temos que
    $3k^2 = k^2 + k^2 + k^2 > k^2 + 2k + 1 = (k+1)^2$
  - Pela propriedade de transitividade
    $3^{k+1} > 3k^2 > (k+1)^2 \rightarrow 3k^2 > (k+1)^2$

Exemplo

- Prove que $\forall n \geq 0 \rightarrow 3^n > n^2$ é verdadeira
  - Se $k > 1$, temos que $3k^2 = k^2 + k^2 + k^2 > k^2 + 2k + 1 = (k+1)^2$
  - Pela propriedade de transitividade $3^{k+1} > 3k^2 > (k+1)^2 \rightarrow 3k^2 > (k+1)^2$
  - Precisa demonstrar que para $n > 1$ que a proposição $3n^2 > (n+1)^2$ é verdadeira

Exemplo

- Prove que $\forall n \geq 1 \rightarrow 3n^2 > (n+1)^2$ é verdadeira
  - Caso base: $n = 2 \rightarrow 3 \times 2^2 \overset{?}{>} (2+1)^2 \rightarrow 12 \overset{V}{>} 9$
  - Hipótese: $\forall k > 2 \rightarrow 3k^2 > (k+1)^2$
  - Tese: $3(k+1)^2 > [(k+1)+1]^2$

$$
\begin{aligned}
3\,(k+1)^2 &\overset{?}{>} [(k+1)+1]^2 \\
3k^2 + 6k + 3 &\overset{?}{>} k^2 + 4k + 4 \\
3k^2 + 6k + 3 &\overset{?}{>} (k+1)^2 + 2k + 3
\end{aligned}
$$

Exemplo

- Prove que $\forall n \geq 1 \rightarrow 3n^2 > (n+1)^2$ é verdadeira
  - Caso base: $n = 2 \rightarrow 3 \times 2^2 \overset{?}{>} (2+1)^2 \rightarrow 12 \overset{V}{>} 9$
  - Hipótese: $\forall k > 2 \rightarrow 3k^2 > (k+1)^2$
  - Tese: $3(k+1)^2 > [(k+1)+1]^2$

$$\begin{aligned} 3\,(k+1)^2 &\overset{?}{>} [(k+1)+1]^2 \\ 3k^2+6k+3 &\overset{?}{>} k^2+4k+4 \\ 3k^2+6k+3 &\overset{?}{>} (k+1)^2+2k+3 \end{aligned}$$

Assumindo a hipótese $3k^2 > (k+1)^2$, é preciso demonstrar que $6k+3 \geq 2k+3$

$$6k+3 \overset{?}{\geq} 2k+3$$

Exemplo

- Prove que $\forall n \geq 1 \rightarrow 3n^2 > (n+1)^2$ é verdadeira
  - Caso base: $n = 2 \rightarrow 3 \times 2^2 \overset{?}{>} (2+1)^2 \rightarrow 12 \overset{V}{>} 9$
  - Hipótese: $\forall k > 2 \rightarrow 3k^2 > (k+1)^2$
  - Tese: $3(k+1)^2 > [(k+1)+1]^2$

$$\begin{aligned} 3\,(k+1)^2 &\overset{?}{>} [(k+1)+1]^2 \\ 3k^2+6k+3 &\overset{?}{>} k^2+4k+4 \\ 3k^2+6k+3 &\overset{?}{>} (k+1)^2+2k+3 \end{aligned}$$

Assumindo a hipótese $3k^2 > (k+1)^2$, é preciso demonstrar que $6k+3 \geq 2k+3$

$$\begin{aligned} 6k+3 &\overset{?}{\geq} 2k+3 \\ 3k &\overset{V}{\geq} k \end{aligned}$$

Exemplo

- Prove que $\forall n \geq 0 \rightarrow 3^n > n^2$ é verdadeira
  - Foi demonstrado que $3n^2 > (n+1)^2$ para $n > 1$

Exemplo

- Prove que $\forall n \geq 0 \rightarrow 3^n > n^2$ é verdadeira
  - Foi demonstrado que $3n^2 > (n+1)^2$ para $n > 1$
  - Com a hipótese $3^k > k^2 \rightarrow 3^{k+1} > 3k^2$ e aplicando a propriedade de transitividade
    $3^{k+1} > 3k^2 > (k+1)^2 \rightarrow 3^{k+1} > (k+1)^2$ para $n > 1$

Exemplo

- Prove que $\forall n \geq 0 \rightarrow 3^n > n^2$ é verdadeira
  - Foi demonstrado que $3n^2 > (n+1)^2$ para $n > 1$
  - Com a hipótese $3^k > k^2 \rightarrow 3^{k+1} > 3k^2$ e aplicando a propriedade de transitividade
    $3^{k+1} > 3k^2 > (k+1)^2 \rightarrow 3^{k+1} > (k+1)^2$ para $n > 1$

A proposição $\forall n \geq 0 \rightarrow 3^n > n^2$ é verdadeira

Exemplo

- Prove por indução a corretude da implementação recursiva do algoritmo de busca binária

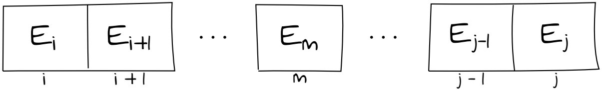
```c
// Padrão de tipos por tamanho
#include <stdint.h>
// Busca binária recursiva
int32_t busca(int32_t x, int32_t* V, int32_t i,
    int32_t j) {
    // Pivô
    int32_t m = (i + j) / 2;
    // Caso base 1
    if(i > j) return -1;
    // Caso base 2
    else if(V[m] == x) return m;
    // Metade inferior
    else if(V[m] > x) return busca(x, V, i, m - 1);
    // Metade superior
    else return busca(x, V, m + 1, j);
}
```

Exemplo

- Prove por indução a corretude da implementação recursiva do algoritmo de busca binária
  - Casos base
    - $P(x, V, i, j) = -1$ se $i > j$
    - $P(x, V, i, j) = y$ se $i = j$ e $V[m = \lfloor (i + j) \div 2 \rfloor] = x$, onde m é o índice do elemento x

Exemplo

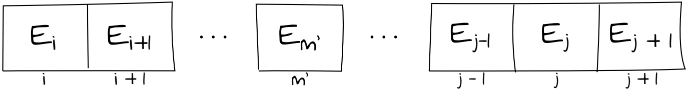
- Prove por indução a corretude da implementação recursiva do algoritmo de busca binária
  - Casos base
    - $P(x, V, i, j) = -1$ se $i > j$
    - $P(x, V, i, j) = y$ se $i = j$ e $V[m = \lfloor (i + j) \div 2 \rfloor] = x$, onde m é o índice do elemento x
  - Hipótese



    - $V[m] = x$, é retornado seu índice $m = \lfloor (i + j) \div 2 \rfloor$
    - $V[m] > x$, x pode estar em $E_i \leq \ldots \leq E_{m-1}$
    - $V[m] < x$, x pode estar em $E_{m+1} \leq \ldots \leq E_j$

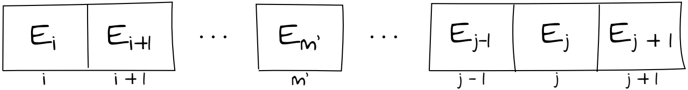
Exemplo

- Prove por indução a corretude da implementação recursiva do algoritmo de busca binária
  - Tese



Exemplo

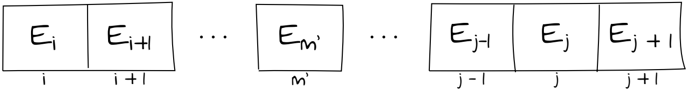
- Prove por indução a corretude da implementação recursiva do algoritmo de busca binária
  - Tese



$$V[m'] = x \to V\left[\left\lfloor \frac{(i+j+1)}{2} \right\rfloor\right] = V\left[\left\lfloor m + \frac{1}{2} \right\rfloor\right] = x$$

Exemplo

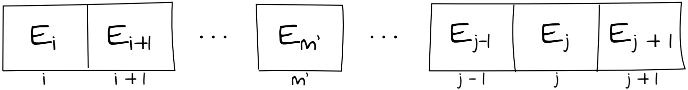
- Prove por indução a corretude da implementação recursiva do algoritmo de busca binária
  - Tese



$$V[m'] > x \rightarrow E_i \leq \ldots \leq E_{m'-1}, \text{ onde } E_{m'-1} < E_{m'}$$

Exemplo

- Prove por indução a corretude da implementação recursiva do algoritmo de busca binária
  - Tese



$$V[m'] < x \rightarrow E_{m'+1} \leq \ldots \leq E_{j+1}, \text{ onde } E_{m'+1} > E_{m'}$$

Exercícios

- Prove por indução matemática que as proposições são verdadeiras, detalhando os casos bases e os passos indutivos
  - $1^2 + 2^2 + \ldots + n^2 = \frac{n(n+1)(2n+1)}{6}$
  - $2^n < n! \quad , \quad n \geq 4$
  - $n^3 + 2n$ é divisível por 3