ANÁLISE DE ALGORITMOS - MAC0338

prof Cristina Fernandes - cris@ime.usp.br

16 de agosto

- operações que o algoritmo faz: número de atribuições ($\leftarrow$)

\# $\leftarrow$ número de atribuições

0 1

1 0

2 $n-1$

3 $n-1$

4 0

5 $\geq 0 \leq \sum_{j=2}^{n} (j-1) = 1 + 2 + \ldots + (n-1) = \frac{n(n-1)}{2}$ = P.A.

6 $\geq 0 \leq \sum_{j=2}^{n} (j-1) = 1 + 2 + \ldots + (n-1) = \frac{n(n-1)}{2}$

7 $n-1$

8 $n-1$

total $= n^2 + 3n - 3$

n^2 domina os outros termos

a execução de cada linha de código consome 1 unidade de tempo

0 1

1 n

2 $n-1$

3 $n-1$

4 $\geq 1 \leq \sum_{j=2}^{n} j = \frac{n(n+1)}{2}$

5 $\geq 0 \leq \sum_{j=2}^{n} (j-1) = 1 + 2 + \ldots + (n-1) = \frac{n(n-1)}{2}$

6 $\geq 0 \leq \sum_{j=2}^{n} (j-1) = 1 + 2 + \ldots + (n-1) = \frac{n(n-1)}{2}$

7 $n-1$

8 $n-1$

16 de agosto

NOTAÇÃO O

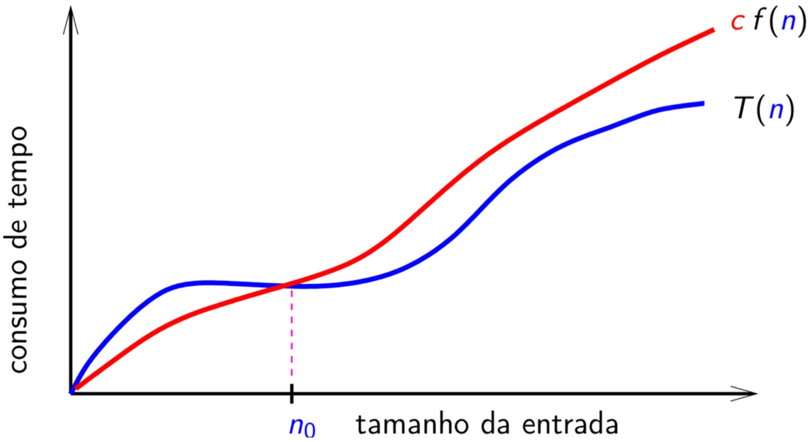
intuitivamente: $O(f(n)) \approx$ funções que não crescem mais rápido que $f(n)$

Definição

Sejam $T(n)$ e $f(n)$ funções dos inteiros nos reais.
Dizemos que $T(n)$ é $O(f(n))$ se
existem constantes positivas c e n_0 tais que

$$T(n) \leq c\, f(n)$$

para todo $n \geq n_0$.



para $n^2 + 99n$ temos c e n_0:

$$n^2 + 99n \leq \underbrace{100}_{c}\, n^2$$

$$\forall n \geq \underbrace{1}_{n_0}$$

18 de agosto

NOTAÇÃO O: delimitação superior

NOTAÇÃO Ω: delimitação inferior

NOTAÇÃO Θ: quando O e Ω possuem a mesma ordem

comparação	comparação assintótica
$T(n) \leq f(n)$	$T(n)$ é $O(f(n))$
$T(n) \geq f(n)$	$T(n)$ é $\Omega(f(n))$
$T(n) = f(n)$	$T(n)$ é $\Theta(f(n))$

24 de agosto de 2022

ALGORITMO ESTÁVEL: se na entrada $A[i] = A[j]$ para $i < j$.

- a partir de uma ordenação anterior, ao ordenar com outro critério a ordenação anterior ainda existe. (confuse)
- exemplo: desejamos ordenar uma lista alfabética de acordo com a idade. para pessoas com a mesma idade prevalece a ordem alfabética.

```
INTERCALA (A, p, q, r)
 0  ▷ B[p . . r] é um vetor auxiliar
 1  para i ← p até q faça
 2      B[i] ← A[i]
 3  para j ← q + 1 até r faça
 4      B[r + q + 1 − j] ← A[j]
 5  i ← p
 6  j ← r
 7  para k ← p até r faça
 8      se B[i] ≤ B[j] e i ≤ q          ▷ alteração
 9          então A[k] ← B[i]
10                i ← i + 1
11          senão A[k] ← B[j]
12                j ← j − 1
```

$n := r - p + 1$

linha	consumo de tempo
1	$1 + \lceil n/2 \rceil = \Theta(n)$
2	$\lceil n/2 \rceil = \Theta(n)$
3	$1 + \lfloor n/2 \rfloor = \Theta(n)$
4	$\lfloor n/2 \rfloor = \Theta(n)$
5	$1 \quad \Theta(n)$
6	$n+1 \quad \Theta(n)$
7	$n \quad \Theta(n)$
8	$\lceil n/2 \rceil \quad \Theta(n)$
9 - 12	$\lfloor n/2 \rfloor \quad \Theta(n)$

$\Theta(n)$

24 de agosto de 2022

$$T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n)$$
$$= T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n$$
$$= T(n/2) + T(n/2) + n$$

n é potência de 2

$T(1) = 1$

expansão

$$= 2T(n/2) + n$$
$$= 2(2T(n/4) + \frac{n}{2}) + n$$
$$= 4T\left(\frac{n}{4}\right) + 2n$$
$$= 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2n$$
$$= 8T\left(\frac{n}{8}\right) + 3n$$
$$= 8\left(2T\left(\frac{n}{16}\right) + \frac{n}{8}\right) + 3n$$
$$= 16T\left(\frac{n}{16}\right) + 4n$$
$$\vdots$$
$$= 2^K T\left(\frac{n}{2^K}\right) + Kn \overset{K = \lg n}{=} n \cdot 1 + n\lg n$$

$$T(n) = n + n\lg n$$
$$T(1) = 1$$

verificação:

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

hip. indução
$$= 2\left(\frac{n}{2} + \frac{n}{2}\lg\frac{n}{2}\right) + n$$
$$= n\lg\frac{n}{2} + 2n$$
$$= n(\lg n - 1)2n = n\lg n + n$$

$T(m) \geq T(n)$ para $m \geq n$

$$2^K \leq m < 2^{K+1}$$

$$2^K + 2^K K = T(2^K) \leq T(m) \leq T(2^{K+1}) = 2^{K+1} + 2^{K+1}(K+1)$$

$$K \leq \lg m < K+1$$

$$T(m) \geq 2^K + 2^K K > 2^{(\lg m - 1)} + 2^{(\lg m - 1)}(\lg m - 1) = \frac{m}{2}\lg m$$

$$T(m) \leq 2^{K+1} + 2^{K+1}(K+1) \leq 2^{\lg m + 1} + 2^{\lg m + 1}(\lg m + 1) = 4m + 2m\lg m$$

$$T(m) = \Theta(m\lg m)$$

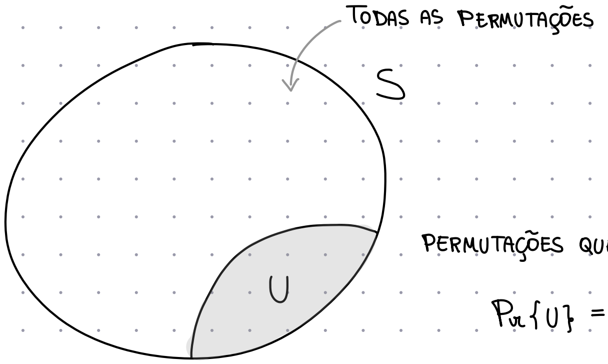
31 de agosto de 2022

ANÁLISE PROBABILÍSTICA

- espaço de probabilidade discreto

S = conjunto finito

(eventos elementares)



PERMUTAÇÕES QUE TERMINAM COM n

$$\Pr\{U\} = \frac{(n-1)!}{n!}$$

- Esperança $E[X]$ de uma variável aleatória
  - linearidade da esperança

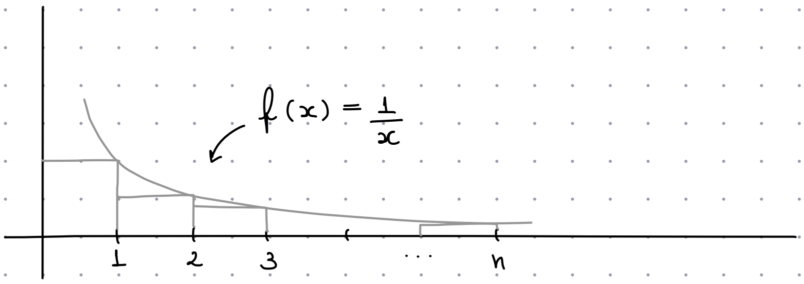
$$\begin{aligned} E[\alpha X + Y] &= \sum_{s \in S} (\alpha X + Y)(s) \Pr\{s\} \\ &= \sum_{s \in S} (\alpha X(s) + Y(s)) \Pr\{s\} \\ &= \sum_{s \in S} X(s) \Pr\{s\} + \sum_{s \in S} Y(s) \Pr\{s\} \end{aligned}$$

- Z = variável aleatória binária

$$\begin{aligned} E[Z] &= 0 \cdot \Pr\{Z=0\} + 1 \cdot \Pr\{Z=1\} \\ &= \Pr\{Z=1\} \end{aligned}$$

$$E[X_i] = \Pr\{X_i = 1\} = \frac{1}{i}$$

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$ = número harmônico



$$\begin{aligned} H_n &= 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} \\ &\leq 1 + \int_1^n \frac{1}{x}\,dx = 1 + \ln \Big|_1^n = 1 + \ln n \end{aligned}$$

$$H_n > \int_1^{n+1} \frac{1}{x}\,dx = \ln(n+1)$$

31 de agosto de 2022

QUICKSORT

- probabilidade do 1 ser comparado com o 100 em uma permutação $= \frac{2}{100}$
- números que estão longe tem menos probabilidade de serem comparados

$$X_{ab} = \begin{cases} 1 \text{ se } a \text{ e } b \text{ forem comparados} \\ 0, \text{ c.c.} \end{cases}$$

$$E[X_{ab}] = \Pr\{X_{ab} = 1\} = \frac{2}{b-a+1}$$

$$E[X] = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} E[X_{ab}] = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \frac{2}{b-a+1} = 2 \sum_{a=1}^{n-1} \boxed{\sum_{t=2}^{n-a+1} \frac{1}{t}} < 2 \sum_{a=1}^{n-1} H_n =$$

$$= 2(n-1) H_n \leq 2(n-1)(1 + \ln n) = O(n \lg n)$$

$$\frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n-a+1} < H_n \leq 1 + \ln n$$

02 de setembro de 2022 - aula 6

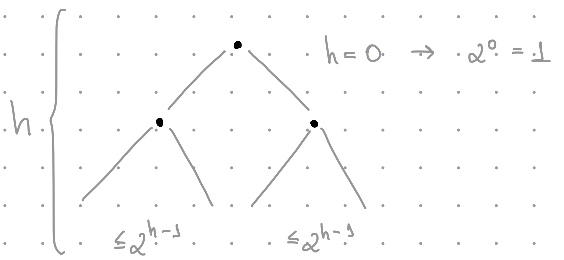
PARTICIONE - SELECT ALEATÓRIO

- número de comparações
  - exemplo: 30 e 70
  - não vão ser comparados se o pivô está entre os dois ou maior que 70

EXERCÍCIO: refaça os cálculos para um K arbitrário

- quebrar a somatória que tem relação com K (três somatórias)
  - dá uma função linear
  - pedaços como: a e b que são menores que K...
  - importante para a prova

- não existe um algoritmo de ordenação baseado em comparação (de propósito geral [os elementos são comparáveis]) que no pior caso consuma n log n.
- árvore de decisões
  - para qualquer algoritmo de ordenação
  - número de folhas: pelo menos o nº de permutações
  - altura da folha: nº de comparações
  - altura da árvore: nº de comparações do pior caso
- árvore binária de altura h
  - no máximo 2^h folhas



- a altura da árvore de decisão tem que ser pelo menos $\frac{n}{2} \lg n$

$$h \geq \lg(n!) \geq \lg \sqrt{n^n} = \lg n^{\frac{n}{2}} = \frac{n}{2} \lg n$$