

SIN5013 - Análise de Algoritmos e Estruturas de Dados

Método para a análise de algoritmos

Prof. Flávio Luiz Coutinho
flcoutinho@usp.br

Método para determinar a função $T(n)$

- $T(n)$: função que expressa o consumo de algum recurso (usualmente o tempo) necessário para a execução de um algoritmo, que processa uma entrada de tamanho n .
- No processo de análise de um algoritmo, o primeiro passo envolve identificar o parâmetro que representa o tamanho do problema (muitas vezes há apenas um parâmetro que usualmente chamamos de n , mas nem sempre é esse o caso).

Método para determinar a função $T(n)$

Exemplo em que mais de um parâmetro é necessário:

- Multiplicação de matrizes: $(m, n) \times (n, p) = (m, p)$
- Quantidade total de multiplicações feitas: $(m * p) * n$
- Quantidade de trabalho depende de 3 parâmetros: m , n e p
- Logo, não dá pra modelar a função que determina a quantidade de multiplicações necessárias apenas pelo total de elementos:

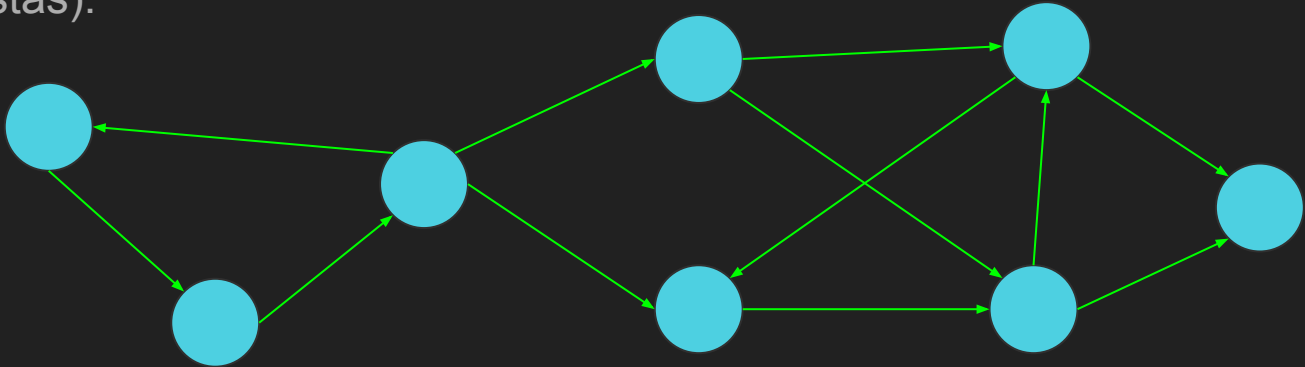
$$(4, 4) \times (4, 4) \text{ ---> } \#mul = 4 * 4 * 4 = 64$$

$$(8, 2) \times (2, 8) \text{ ---> } \#mul = 8 * 2 * 8 = 128$$

$$(2, 8) \times (8, 2) \text{ ---> } \#mul = 2 * 8 * 2 = 32$$

Método para determinar a função $T(n)$

Outro exemplo em que é necessário mais de um parâmetro relacionado ao tamanho da entrada (ou tamanho do problema) seriam os algoritmos que manipulam **grafos** (estruturas de dados que representam um conjunto de vértices conectados por um conjunto de arestas).



Para tais algoritmos, o volume de trabalho a ser feito (e consequentemente o tempo de execução) será definido tanto pela quantidade de vértices, quanto pela quantidade de arestas.

Método para determinar a função $T(n)$

- **Segundo passo:** para cada linha do algoritmo (não importa a linguagem ou mesmo se é pseudo-código), atribuir um **custo abstrato** de tempo (ou seja, o tempo que a linha em questão leva para ser executada apenas 1 vez).
- Por que um custo abstrato (e não real)?
 - Custo real dependente do ambiente em particular (arquitetura, SO, etc).
 - Difícil de determinar (seja experimentalmente, seja teoricamente).
 - Trabalhar com custo abstrato ou real é indiferente, uma vez que nosso objetivo é compreender o comportamento da função $T(n)$ e não determiná-la de forma exata (ou seja queremos determinar sua **complexidade assintótica**).

Método para determinar a função $T(n)$

- Cuidado com linhas que não possuem custo constante, como chamadas de funções e, em especial, chamadas recursivas!!!

Método para determinar a função $T(n)$

- **Terceiro passo:** determinar quantas vezes cada linha do algoritmo (que já possui um custo abstrato atribuído) é executada, para resolver um problema de tamanho n (assumindo que o tamanho do problema em questão é parametrizado por apenas uma variável).
- Provavelmente esta é a etapa mais crítica (nem sempre é fácil determinar o número de repetições de uma linha do algoritmo).

Método para determinar a função $T(n)$

- Quarto passo: determinar a função $T(n)$:

$$T(n) = \sum (\text{custo da linha } i) * (\text{quantidade de execuções da linha } i)$$

- Quinto passo: simplificações:
 - agrupar constantes
 - desconsiderar termos de menor ordem
 - desconsiderar coeficiente do termo de maior ordem

Método para determinar a função $T(n)$

Resumo:

- identificar parâmetro que define tamanho do problema
- atribuir custo abstrato por linha
- determinar quantidade de vezes que cada linha executa
- determinar $T(n)$
- simplificações

Resumo de simplificações adotadas em todo o processo:

- custos abstratos
- agrupamento de constantes
- termos de menor ordem ignorados
- coeficiente do termo de maior ordem ignorado

Método para determinar a função $T(n)$

Resumo:

- identificar parâmetro que define tamanho do problema
- atribuir custo abstrato por linha
- determinar quantidade de vezes que cada linha executa
- determinar $T(n)$
- simplificações

Resumo de simplificações adotadas em todo o processo:

- custos abstratos
- agrupamento de constantes
- termos de menor ordem ignorados
- coeficiente do termo de maior ordem ignorado

Com a experiência, é normal a gente pular algumas etapas neste processo!

Método para determinar a função $T(n)$

Exemplo:

- função **max**: dado um vetor (*array*) de valores inteiros, determinar e devolver qual o maior valor.
- Versões:
 - Iterativa
 - Recursiva com resolução de 1 subproblema de tamanho $(n - 1)$
 - Recursiva com resolução de 2 subproblemas de tamanho $(n / 2)$.