

Trilha de Teoria da Computação

Para receber um certificado dessa trilha, o aluno deve cursar

- ▶ as obrigatórias de pelo menos 2 módulos e
- ▶ pelo menos 7 disciplinas da trilha.

Trilha de Teoria da Computação

Para receber um certificado dessa trilha, o aluno deve cursar

- ▶ as obrigatórias de pelo menos 2 módulos e
- ▶ pelo menos 7 disciplinas da trilha.

Três módulos:

- ▶ Matemática discreta
- ▶ Algoritmos
- ▶ Otimização

Trilha de Teoria da Computação

Para receber um certificado dessa trilha, o aluno deve cursar

- ▶ as obrigatórias de pelo menos 2 módulos e
- ▶ pelo menos 7 disciplinas da trilha.

Três módulos:

- ▶ Matemática discreta
- ▶ Algoritmos
- ▶ Otimização

Hoje: **Módulo Algoritmos**

Escalonamento de máquinas idênticas

Dados: m máquinas

t tarefas

duração $d[i]$ da tarefa i ($i = 1, \dots, t$)

Escalonamento de máquinas idênticas

Dados: m máquinas

t tarefas

duração $d[i]$ da tarefa i ($i = 1, \dots, t$)

um escalonamento é

uma partição $\{M[1], \dots, M[m]\}$ de $\{1, \dots, t\}$

Escalonamento de máquinas idênticas

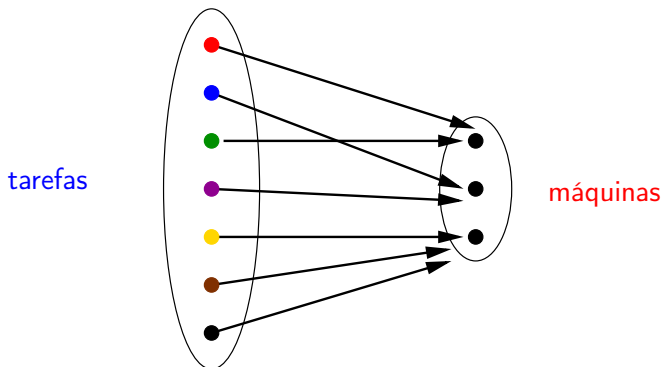
Dados: m máquinas

t tarefas

duração $d[i]$ da tarefa i ($i = 1, \dots, t$)

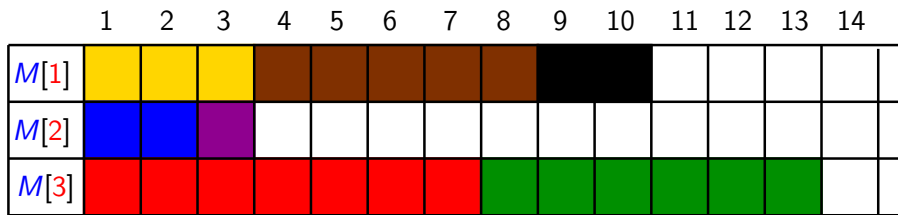
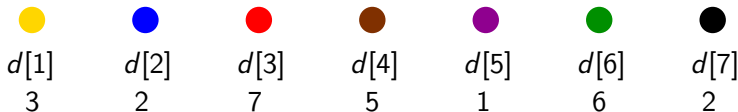
um escalonamento é

uma partição $\{M[1], \dots, M[m]\}$ de $\{1, \dots, t\}$



Exemplo 1

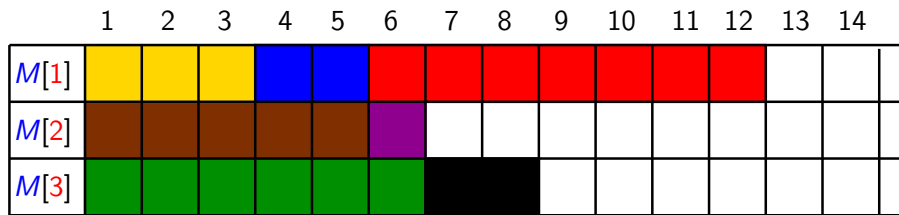
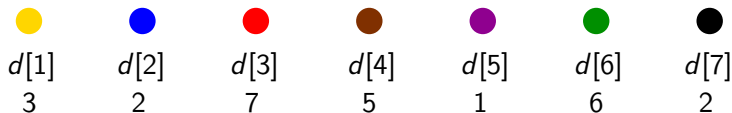
$$m = 3 \quad t = 7$$



$\{\{1, 4, 7\}, \{2, 5\}, \{3, 6\}\} \Rightarrow \text{Tempo de conclusão} = 13$

Exemplo 2

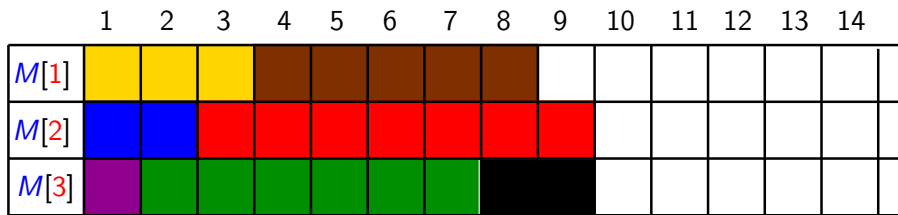
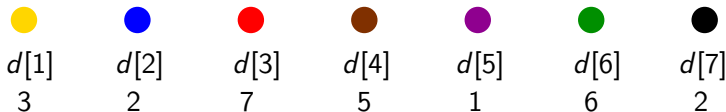
$$m = 3 \quad t = 7$$



$\{\{1, 2, 3\}, \{4, 5\}, \{6, 7\}\} \Rightarrow$ Tempo de conclusão = 12

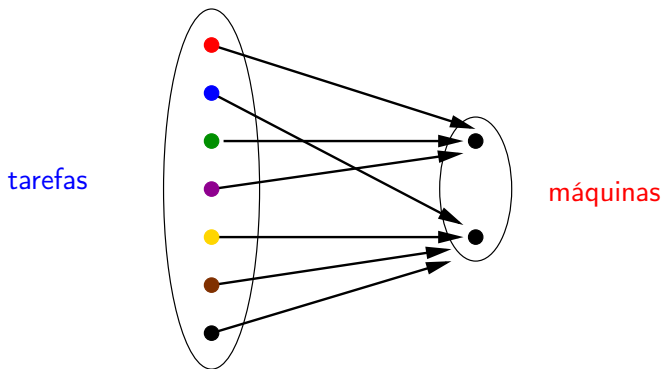
Problema

Encontrar um escalonamento com tempo de conclusão **mínimo**.



$\{\{1, 4\}, \{2, 3\}, \{5, 6, 7\}\} \Rightarrow$ Tempo de conclusão = 9

NP-difícil mesmo para $m = 2$



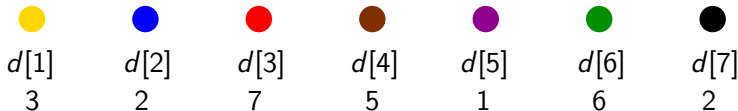
Algoritmo: testa todo $M[1] \subseteq \{1, \dots, t\}$ e escolhe melhor

2^t subconjuntos $\Rightarrow$ exponencial

NP-difícil $\Rightarrow$ é improvável que exista algoritmo polinomial que resolva o problema (se existir, $P = NP$)

Algoritmo de Graham

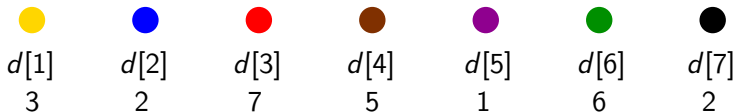
Atribui, uma a uma, cada tarefa à **máquina** **menos** ocupada.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$M[1]$														
$M[2]$														
$M[3]$														

Algoritmo de Graham

Atribui, uma a uma, cada tarefa à **máquina** **menos** ocupada.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$M[1]$														
$M[2]$														
$M[3]$														

Algoritmo de Graham

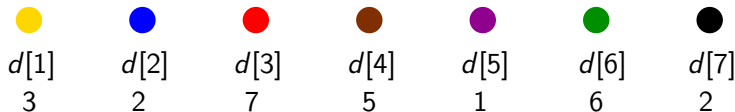
Atribui, uma a uma, cada tarefa à máquina menos ocupada.

						
$d[1]$	$d[2]$	$d[3]$	$d[4]$	$d[5]$	$d[6]$	$d[7]$
3	2	7	5	1	6	2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$M[1]$														
$M[2]$														
$M[3]$														

Algoritmo de Graham

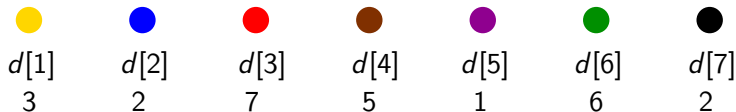
Atribui, uma a uma, cada tarefa à **máquina** **menos** ocupada.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$M[1]$														
$M[2]$														
$M[3]$														

Algoritmo de Graham

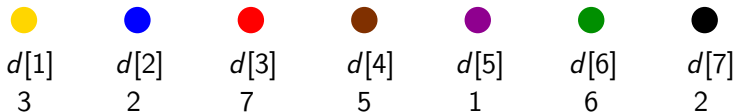
Atribui, uma a uma, cada tarefa à **máquina** **menos** ocupada.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$M[1]$														
$M[2]$														
$M[3]$														

Algoritmo de Graham

Atribui, uma a uma, cada tarefa à **máquina** **menos** ocupada.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$M[1]$														
$M[2]$														
$M[3]$														

Algoritmo de Graham

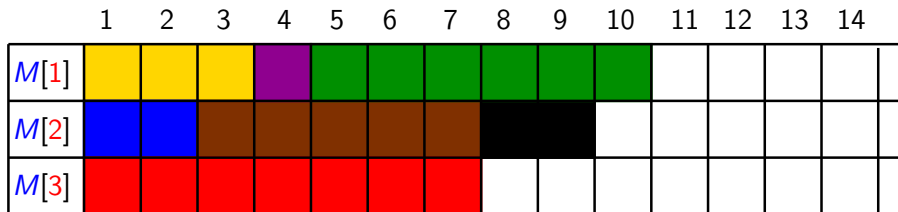
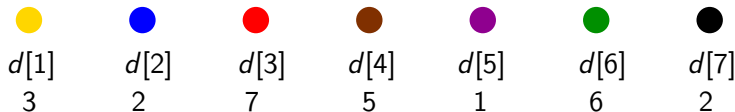
Atribui, uma a uma, cada tarefa à máquina menos ocupada.

						
$d[1]$	$d[2]$	$d[3]$	$d[4]$	$d[5]$	$d[6]$	$d[7]$
3	2	7	5	1	6	2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$M[1]$														
$M[2]$														
$M[3]$														

Algoritmo de Graham

Atribui, uma a uma, cada tarefa à **máquina menos** ocupada.



Escalonamento-Graham

Recebe números inteiros positivos m e t e um vetor $d[1..t]$ e devolve um escalonamento de $\{1, \dots, t\}$ em m máquinas.

ESCALONAMENTO-GRAHAM (m, t, d)

```
1  para  $j \leftarrow 1$  até  $m$  faça
2       $M[j] \leftarrow \emptyset$ 
3       $T[j] \leftarrow 0$ 
4  para  $i \leftarrow 1$  até  $t$  faça
5      seja  $k$  tal que  $T[k]$  é mínimo
6       $M[k] \leftarrow M[k] \cup \{i\}$ 
7       $T[k] \leftarrow T[k] + d[i]$ 
8  devolva  $\{M[1], \dots, M[m]\}$ 
```

Delimitações para OPT

OPT = menor tempo de conclusão de um escalonamento

- ▶ Duração da tarefa mais longa:

$$\text{OPT} \geq \max\{d[1], d[2], \dots, d[t]\}$$

- ▶ Distribuição balanceada:

$$\text{OPT} \geq \frac{d[1] + d[2] + \dots + d[t]}{m}$$

Qualidade do escalonamento

	1	2	3	4	...	T					T_G			
$M[1]$														
$\vdots$														
$M[j]$														
$\vdots$														
$M[m]$														

$$\begin{aligned} T_G &= T + d[t] \\ &\leq \frac{d[1] + \dots + d[t]}{m} + d[t] \\ &\leq \frac{d[1] + \dots + d[t]}{m} + \max\{d[1], \dots, d[t]\} \\ &\leq \text{OPT} + \text{OPT} = 2 \text{OPT} \end{aligned}$$

Conclusão

ESCALONAMENTO-GRAHAM (m, t, d)

```
1  para  $j \leftarrow 1$  até  $m$  faça
2       $M[j] \leftarrow \emptyset$ 
3       $T[j] \leftarrow 0$ 
4  para  $i \leftarrow 1$  até  $t$  faça
5      seja  $k$  tal que  $T[k]$  é mínimo
6       $M[k] \leftarrow M[k] \cup \{i\}$ 
7       $T[k] \leftarrow T[k] + d[i]$ 
8  devolva  $\{M[1], \dots, M[m]\}$ 
```

O algoritmo ESCALONAMENTO-GRAHAM é uma 2-aproximação.

Módulo Matemática Discreta

- ▶ MAT0206 Análise Real
- ▶ MAT0264 Anéis e corpos
- ▶ MAC0320 Introdução à Teoria dos Grafos

Módulo Matemática Discreta

- ▶ MAT0206 Análise Real
- ▶ MAT0264 Aneis e corpos
- ▶ MAC0320 Introdução à Teoria dos Grafos

- ▶ MAT0311 Cálculo Diferencial e Integral V
- ▶ MAT0225 Funções Analíticas
- ▶ MAT0313 Álgebra III
- ▶ MAT0234 Medida e Integração
- ▶ MAE0221 Probabilidade I
- ▶ MAE0224 Probabilidade II
- ▶ MAE0228 Noções de Probabilidade e Processos Estocásticos
- ▶ MAE0326 Aplicações de Processos Estocásticos
- ▶ MAC0313 Autômatos, Computabilidade e Complexidade
- ▶ MAC0436 Tópicos de Matemática Discreta

Módulo Algoritmos

- ▶ MAC0328 Algoritmos em Grafos
- ▶ MAC0414 Autômatos, Computabilidade e Complexidade

Módulo Algoritmos

- ▶ MAC0328 Algoritmos em Grafos
- ▶ MAC0414 Autômatos, Computabilidade e Complexidade
- ▶ MAC0325 Otimização Combinatória
- ▶ MAC0327 Desafios de Programação
- ▶ MAC0331 Geometria Computacional
- ▶ MAC0450 Algoritmos de Aproximação
- ▶ MAC0336 Criptografia para Segurança de Dados
- ▶ MAC0465 Biologia Computacional
- ▶ MAC0466 Teoria dos Jogos Algorítmica

Módulo Otimização

- ▶ MAC0315 Programação Linear
- ▶ MAC0325 Otimização Combinatória

Módulo Otimização

- ▶ MAC0315 Programação Linear
- ▶ MAC0325 Otimização Combinatória
- ▶ MAC0300 Métodos Numéricos da Álgebra Linear
- ▶ MAC0??? Programação Inteira (a ser criada)
- ▶ MAC0427 Programação Não Linear
- ▶ MAC0419 Métodos de Otimização em Finanças
- ▶ MAC0450 Algoritmos de Aproximação
- ▶ MAC0452 Tópicos de Otimização Combinatória
- ▶ MAC0418 Tópicos Especiais de Programação Matemática
- ▶ MAC0461 Introdução ao Escalonamento e Aplicações

MAC0328 Algoritmos em Grafos

Objetivos: Estudar algoritmos para problemas fundamentais em grafos.

MAC0328 Algoritmos em Grafos

Objetivos: Estudar algoritmos para problemas fundamentais em grafos.

Programa:

- ▶ Conexão de grafos: componentes, grafos biconexos.
- ▶ Digrafos fortemente conexos (alg. Kosaraju-Sharir, alg. Tarjan).
- ▶ Emparelhamentos máximos em grafos bipartidos.
- ▶ Emparelhamentos em grafos arbitrários (alg. Edmonds).
- ▶ Fluxo máximo (alg. Ford-Fulkerson).
- ▶ Coloração de vértices.
- ▶ Circuitos hamiltonianos.
- ▶ Tópicos opcionais:
link analysis, network analysis, redes aleatórias.

MAC0328 Algoritmos em Grafos

Objetivos: Estudar algoritmos para problemas fundamentais em grafos.

Programa:

- ▶ Conexão de grafos: componentes, grafos biconexos.
- ▶ Digrafos fortemente conexos (alg. Kosaraju-Sharir, alg. Tarjan).
- ▶ Emparelhamentos máximos em grafos bipartidos.
- ▶ Emparelhamentos em grafos arbitrários (alg. Edmonds).
- ▶ Fluxo máximo (alg. Ford-Fulkerson).
- ▶ Coloração de vértices.
- ▶ Circuitos hamiltonianos.
- ▶ Tópicos opcionais:
link analysis, network analysis, redes aleatórias.

Bibliografia principal:

Sedgewick, K. Wayne, Algorithms, 4a ed., 2011.

MAC0414 Autômatos, Computabilidade e Complexidade

Objetivos: Estudo de vários formalismos para computação e algoritmos e as limitações de certas formas de computação.

MAC0414 Autômatos, Computabilidade e Complexidade

Objetivos: Estudo de vários formalismos para computação e algoritmos e as limitações de certas formas de computação.

Programa:

- ▶ Palavras, linguagens, operações sobre linguagens.
- ▶ Linguagens regulares.
- ▶ Autômatos finitos determinísticos e não determinísticos.
- ▶ Teorema de Kleene.
- ▶ Autômatos reduzidos.
- ▶ Modelos de computação; máquinas de Turing.
- ▶ Tese de Church.
- ▶ Redutibilidade e problemas indecidíveis.
- ▶ Complexidade, problemas decidíveis em tempo polinomial.
- ▶ Não-determinismo versus determinismo.
- ▶ Redutibilidade e problemas NP-completos.
- ▶ Teorema de Cook-Levin.

MAC0414 Autômatos, Computabilidade e Complexidade

Objetivos: Estudo de vários formalismos para computação e algoritmos e as limitações de certas formas de computação.

Programa:

- ▶ Palavras, linguagens, operações sobre linguagens.
- ▶ Linguagens regulares.
- ▶ Autômatos finitos determinísticos e não determinísticos.
- ▶ Teorema de Kleene.
- ▶ Autômatos reduzidos.
- ▶ Modelos de computação; máquinas de Turing.
- ▶ Tese de Church.
- ▶ Redutibilidade e problemas indecidíveis.
- ▶ Complexidade, problemas decidíveis em tempo polinomial.
- ▶ Não-determinismo versus determinismo.
- ▶ Redutibilidade e problemas NP-completos.
- ▶ Teorema de Cook-Levin.

Bibliografia principal:

Sipser, Introduction to the Theory of Computation, 3a ed., 2012.

Módulo Algoritmos

Demais disciplinas:

- ▶ MAC0325 Otimização Combinatória
- ▶ MAC0331 Geometria Computacional
- ▶ MAC0450 Algoritmos de Aproximação
- ▶ MAC0327 Desafios de Programação
- ▶ MAC0336 Criptografia para Segurança de Dados
- ▶ MAC0465 Biologia Computacional
- ▶ MAC0466 Teoria dos Jogos Algorítmica