Análise de Algoritmos

Parte destes slides são adaptações de slides

do Prof. Paulo Feofiloff e do Prof. José Coelho de Pina.

Mais um problema de escalonamento

d_i: deadline da tarefa i

t_i: tempo de processamento da tarefa i

Escalonamento: permutação π, onde
$\pi(i)$ denota a posição da tarefa i na ordem de execução.

Mais um problema de escalonamento

d_i: deadline da tarefa i

t_i: tempo de processamento da tarefa i

Escalonamento: permutação π, onde
$\pi(i)$ denota a posição da tarefa i na ordem de execução.

Dado π, o instante de início da tarefa i é

$$s_i = \sum_{j=1}^{\pi(i)-1} t_{\pi^{-1}(j)},$$

ou seja, é a soma das durações das tarefas anteriores a i.

Mais um problema de escalonamento

d_i: deadline da tarefa i

t_i: tempo de processamento da tarefa i

Escalonamento: permutação π, onde
$\pi(i)$ denota a posição da tarefa i na ordem de execução.

Dado π, o instante de início da tarefa i é

$$s_i \;=\; \sum_{j=1}^{\pi(i)-1} t_{\pi^{-1}(j)},$$

ou seja, é a soma das durações das tarefas anteriores a i.

Instante de término: $f_i = s_i + t_i$.

Mais um problema de escalonamento

d_i: deadline da tarefa i

t_i: tempo de processamento da tarefa i

Escalonamento: permutação π, onde
$\pi(i)$ denota a posição da tarefa i na ordem de execução.

Dado π, o instante de início da tarefa i é

$$s_i = \sum_{j=1}^{\pi(i)-1} t_{\pi^{-1}(j)},$$

ou seja, é a soma das durações das tarefas anteriores a i.

Instante de término: $f_i = s_i + t_i$.

Atraso ℓ_i: 0 se $f_i \leq d_i$ e $d_i - f_i$ c.c.

Mais um problema de escalonamento

d_i: deadline da tarefa i
t_i: tempo de processamento da tarefa i

Dado escalonamento π,
s_i: início do processamento da tarefa i
f_i: fim do processamento da tarefa i
ℓ_i: atraso da tarefa i

Mais um problema de escalonamento

d_i: deadline da tarefa i
t_i: tempo de processamento da tarefa i

Dado escalonamento π,
s_i: início do processamento da tarefa i
f_i: fim do processamento da tarefa i
ℓ_i: atraso da tarefa i

Problema: Dados d e t, encontrar π cujo atraso máximo é mínimo.

Mais um problema de escalonamento

d_i: deadline da tarefa i
t_i: tempo de processamento da tarefa i

Dado escalonamento π,
s_i: início do processamento da tarefa i
f_i: fim do processamento da tarefa i
ℓ_i: atraso da tarefa i

Problema: Dados d e t, encontrar π cujo atraso máximo é mínimo.

Exemplo: $t_1 = 1$ e $d_1 = 2$, $t_2 = 2$ e $d_2 = 4$, $t_3 = 3$ e $d_3 = 6$.

Mais um problema de escalonamento

d_i: deadline da tarefa i
t_i: tempo de processamento da tarefa i

Dado escalonamento π,
s_i: início do processamento da tarefa i
f_i: fim do processamento da tarefa i
ℓ_i: atraso da tarefa i

Problema: Dados d e t, encontrar π cujo atraso máximo é mínimo.

Exemplo: $t_1 = 1$ e $d_1 = 2$, $t_2 = 2$ e $d_2 = 4$, $t_3 = 3$ e $d_3 = 6$.

Escalonamento com atraso mínimo: $(1, 2, 3)$

Atrasos: $\ell_1 = \ell_2 = \ell_3 = 0$.

Possíveis critérios gulosos

- SPT - shortest processing time (menor t_i primeiro)

Funciona?

Possíveis critérios gulosos

- SPT - shortest processing time (menor t_i primeiro)

Funciona?

Não...
Exemplo: $d_1 = 10$ e $t_1 = 1$, $d_2 = 8$ e $t_2 = 8$.

Possíveis critérios gulosos

- SPT - shortest processing time (menor t_i primeiro)

Funciona?

Não...
Exemplo: $d_1 = 10$ e $t_1 = 1$, $d_2 = 8$ e $t_2 = 8$.

- LST - least slack time (menor $d_i - t_i$ primeiro)

Funciona?

Possíveis critérios gulosos

- SPT - shortest processing time (menor t_i primeiro)

Funciona?

Não...
Exemplo: $d_1 = 10$ e $t_1 = 1$, $d_2 = 8$ e $t_2 = 8$.

- LST - least slack time (menor $d_i - t_i$ primeiro)

Funciona?

Também não...
Exemplo: $d_1 = 2$ e $t_1 = 1$, $d_2 = 10$ e $t_2 = 10$.

Possíveis critérios gulosos

- SPT - shortest processing time (menor t_i primeiro)

Funciona?

Não...
Exemplo: $d_1 = 10$ e $t_1 = 1$, $d_2 = 8$ e $t_2 = 8$.

- LST - least slack time (menor $d_i - t_i$ primeiro)

Funciona?

Também não...
Exemplo: $d_1 = 2$ e $t_1 = 1$, $d_2 = 10$ e $t_2 = 10$.

- EDD - earliest due date (menor d_i primeiro)

Funciona?

Algoritmo resultante

GULOSO (d, t, n)

1 seja π permutação tq $d[\pi^{-1}(1)] \leq \cdots \leq d[\pi^{-1}(n)]$

2 **devolva** π

Nem olhamos o t... Será que isso funciona???

Algoritmo resultante

GULOSO (d, t, n)
1 seja π permutação tq $d[\pi^{-1}(1)] \leq \cdots \leq d[\pi^{-1}(n)]$
2 **devolva** π

Nem olhamos o t... Será que isso funciona???

Note que não há tempo ocioso em uma solução.

Dado um escalonamento, uma inversão é um par de índices i e j tais que $\pi(i) < \pi(j)$ e $d_i > d_j$.

Algoritmo resultante

GULOSO (d, t, n)
1 seja π permutação tq $d[\pi^{-1}(1)] \leq \cdots \leq d[\pi^{-1}(n)]$
2 **devolva** π

Nem olhamos o t... Será que isso funciona???

Note que não há tempo ocioso em uma solução.

Dado um escalonamento, uma inversão é um par de índices i e j tais que $\pi(i) < \pi(j)$ e $d_i > d_j$.

Afirmação 1: Dois escalonamentos sem inversão têm o mesmo atraso máximo.

Algoritmo resultante

GULOSO (d, t, n)
1 seja π permutação tq $d[\pi^{-1}(1)] \leq \cdots \leq d[\pi^{-1}(n)]$
2 **devolva** π

Nem olhamos o t... Será que isso funciona???

Note que não há tempo ocioso em uma solução.

Dado um escalonamento, uma inversão é um par de índices i e j tais que $\pi(i) < \pi(j)$ e $d_i > d_j$.

Afirmação 1: Dois escalonamentos sem inversão têm o mesmo atraso máximo.

Prova feita na aula.

Algoritmo resultante

GULOSO (d, t, n)
1 seja π permutação tq $d[\pi^{-1}(1)] \leq \cdots \leq d[\pi^{-1}(n)]$
2 **devolva** π

Nem olhamos o t... Será que isso funciona???

Note que não há tempo ocioso em uma solução.

Dado um escalonamento, uma inversão é um par de índices i e j tais que $\pi(i) < \pi(j)$ e $d_i > d_j$.

Afirmação 1: Dois escalonamentos sem inversão têm o mesmo atraso máximo.

Afirmação 2: Existe uma solução que não tem nenhuma inversão.

Provas feitas na aula.

Matroides e o método guloso

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.
(subconjuntos de conjuntos em $\mathcal{C}$ estão em $\mathcal{C}$)

Matroides e o método guloso

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

Problema 1: Dado um peso binário w_e para cada e de U, encontrar um conjunto de $\mathcal{C}$ de peso máximo.

Matroides e o método guloso

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

Problema 1: Dado um peso binário w_e para cada e de U, encontrar um conjunto de $\mathcal{C}$ de peso máximo.

```
GULOSO (U, w, C)
    U_1 ← {e ∈ U : w_e = 1}        S ← ∅
    enquanto existe e em U_1 tal que S ∪ {e} ∈ C faça
        S ← S ∪ {e}
    devolva S
```

Matroides e o método guloso

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

Problema 1: Dado um peso binário w_e para cada e de U, encontrar um conjunto de $\mathcal{C}$ de peso máximo.

GULOSO $(U, w, \mathcal{C})$

$U_1 \leftarrow \{e \in U : w_e = 1\}$ $\quad S \leftarrow \emptyset$

enquanto existe e em U_1 tal que $S \cup \{e\} \in \mathcal{C}$ **faça**

$\quad S \leftarrow S \cup \{e\}$

devolva S

GULOSO encontra um conjunto de $\mathcal{C}$ de peso maximal.

$\mathcal{C}$ é um matroide se todo conjunto de $\mathcal{C}$ de peso maximal tem o mesmo tamanho.

Matroides e o método guloso

U: conjunto finito.
$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

Pesos positivos para os elementos de U.

Problema 2: Encontrar um conjunto de $\mathcal{C}$ de peso máximo.

Matroides e o método guloso

U: conjunto finito.
$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

Pesos positivos para os elementos de U.

Problema 2: Encontrar um conjunto de $\mathcal{C}$ de peso máximo.

GULOSO $(U, w, \mathcal{C})$
1 $(e_1, \ldots, e_n) \leftarrow$ ORDENE(U, w) $\triangleright$ ordem dos pesos
2 $S \leftarrow \emptyset$
3 **para** $i \leftarrow 1$ **até** n **faça**
4 se $S \cup \{e_i\} \in \mathcal{C}$
5 **então** $S \leftarrow S \cup \{e_i\}$
6 **devolva** S

Matroides e o método guloso

U: conjunto finito.
$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

Pesos positivos para os elementos de U.

Problema 2: Encontrar um conjunto de $\mathcal{C}$ de peso máximo.

GULOSO $(U, w, \mathcal{C})$
1 $(e_1, \ldots, e_n) \leftarrow$ ORDENE(U, w) ▷ ordem dos pesos
2 $S \leftarrow \emptyset$
3 **para** $i \leftarrow 1$ **até** n **faça**
4 se $S \cup \{e_i\} \in \mathcal{C}$
5 **então** $S \leftarrow S \cup \{e_i\}$
6 **devolva** S

Teorema: Se $\mathcal{C}$ é um matroide, então o algoritmo acima resolve o Problema 2.

Exemplos de matroides

Dado um espaço vetorial, o coleção de todos os conjuntos de vetores LI deste espaço é um matroide.

Exemplos de matroides

Dado um espaço vetorial, o coleção de todos os conjuntos de vetores LI deste espaço é um matroide.

Dado um grafo G, a coleção de arestas de todas as florestas de G é um matroide.

Exemplos de matroides

Dado um espaço vetorial, o coleção de todos os conjuntos de vetores LI deste espaço é um matroide.

Dado um grafo G, a coleção de arestas de todas as florestas de G é um matroide.

Grafo bipartido $G = (U \times V, E)$.
M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.
M_U é um matroide.

Exemplos de matroides

Dado um espaço vetorial, o coleção de todos os conjuntos de vetores LI deste espaço é um matroide.

Dado um grafo G, a coleção de arestas de todas as florestas de G é um matroide.

Grafo bipartido $G = (U \times V, E)$.
M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.
M_U é um matroide.

M_V: a coleção análoga com V no lugar de U.

Claro que M_V também é um matroide.

Exemplos de matroides

Dado um espaço vetorial, o coleção de todos os conjuntos de vetores LI deste espaço é um matroide.

Dado um grafo G, a coleção de arestas de todas as florestas de G é um matroide.

Grafo bipartido $G = (U \times V, E)$.
M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.
M_U é um matroide.

M_V: a coleção análoga com V no lugar de U.

Claro que M_V também é um matroide.

E $M_U \cap M_V$? É ou não é um matroide?

Exemplos de matroides

Grafo bipartido $G = (U \times V, E)$.

M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.

M_V: a coleção analoga com V no lugar de U.

M_U e M_V são matroides.

Exemplos de matroides

Grafo bipartido $G = (U \times V, E)$.

M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.

M_V: a coleção analoga com V no lugar de U.

M_U e M_V são matroides.

O que é $M_U \cap M_V$?

O que é um conjunto da coleção $M_U \cap M_V$ em G?

Exemplos de matroides

Grafo bipartido $G = (U \times V, E)$.

M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.

M_V: a coleção analoga com V no lugar de U.

M_U e M_V são matroides.

O que é $M_U \cap M_V$?

O que é um conjunto da coleção $M_U \cap M_V$ em G?

Cada conjunto de $M_U \cap M_V$ é um emparelhamento em G.

Exemplos de matroides

Grafo bipartido $G = (U \times V, E)$.

M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.

M_V: a coleção analoga com V no lugar de U.

M_U e M_V são matroides.

O que é $M_U \cap M_V$?

O que é um conjunto da coleção $M_U \cap M_V$ em G?

Cada conjunto de $M_U \cap M_V$ é um emparelhamento em G.

Esta coleção é ou não é um matroide?

Exemplos de matroides

Grafo bipartido $G = (U \times V, E)$.

M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.

M_V: a coleção analoga com V no lugar de U.

M_U e M_V são matroides.

O que é $M_U \cap M_V$?

O que é um conjunto da coleção $M_U \cap M_V$ em G?

Cada conjunto de $M_U \cap M_V$ é um emparelhamento em G.

Esta coleção é ou não é um matroide?

Não é... (nem todo emparelhamento maximal é máximo)
Mas é a interseção de dois matroides.

Exemplos de matroides

Grafo bipartido $G = (U \times V, E)$.

M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.

M_V: a coleção analoga com V no lugar de U.

M_U e M_V são matroides.

$M_U \cap M_V$ é a coleção dos emparelhamentos de G.

Esta coleção não é um matroide.

Exemplos de matroides

Grafo bipartido $G = (U \times V, E)$.

M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.

M_V: a coleção analoga com V no lugar de U.

M_U e M_V são matroides.

$M_U \cap M_V$ é a coleção dos emparelhamentos de G.

Esta coleção não é um matroide.

Mas é a interseção de dois matroides.

Exemplos de matroides

Grafo bipartido $G = (U \times V, E)$.

M_U: todos os conjuntos S de arestas de G tq no máximo uma aresta de S é incidente a cada vértice de U.

M_V: a coleção analoga com V no lugar de U.

M_U e M_V são matroides.

$M_U \cap M_V$ é a coleção dos emparelhamentos de G.

Esta coleção não é um matroide.

Mas é a interseção de dois matroides.

Existe algoritmo polinomial para encontrar um conjunto (de peso) máximo na interseção de dois matroides.

Definição alternativa de matroides

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

Definição alternativa de matroides

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

$\mathcal{C}$ é um matroide se, para todo par A, B de conjuntos de $\mathcal{C}$ para os quais $|A| < |B|$, existe $e \in B \setminus A$ tal que $A \cup \{e\} \in \mathcal{C}$.

Definição alternativa de matroides

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

$\mathcal{C}$ é um matroide se, para todo par A, B de conjuntos de $\mathcal{C}$ para os quais $|A| < |B|$, existe $e \in B \setminus A$ tal que $A \cup \{e\} \in \mathcal{C}$.

Esta é a definição do CLRS.

Definição alternativa de matroides

U: conjunto finito arbitrário.

$\mathcal{C}$: família não vazia de subconjuntos de U hereditária.

$\mathcal{C}$ é um matroide se, para todo par A, B de conjuntos de $\mathcal{C}$ para os quais $|A| < |B|$, existe $e \in B \setminus A$ tal que $A \cup \{e\} \in \mathcal{C}$.

Esta é a definição do CLRS.

Exercício: Mostre que esta definição é análoga a anterior.