Análise de Algoritmos

Parte destes slides são adaptações de slides

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

Algoritmos gulosos (*greedy*)

CLRS Cap 16 e KT Cap 4

Algoritmos gulosos

"A *greedy algorithm* starts with a solution to a very small subproblem and augments it successively to a solution for the big problem. The augmentation is done in a "greedy" fashion, that is, paying atention to short-term or local gain, without regard to whether it will lead to a good long-term or global solution. As in real life, greedy algorithms sometimes lead to the best solution, sometimes lead to pretty good solutions, and sometimes lead to lousy solutions. The trick is to determine when to be greedy."

"One thing you will notice about greedy algorithms is that they are usually easy to design, easy to implement, easy to analyse, and they are very fast, but they are *almost always difficult to prove correct*."

I. Parberry, *Problems on Algorithms*, Prentice Hall, 1995.

Algoritmos gulosos

Algoritmo guloso

- procura ótimo local e acaba obtendo ótimo global

Costuma ser

- muito simples e intuitivo
- muito eficiente
- difícil provar que está correto

Problema precisa ter

- subestrutura ótima (como na programação dinâmica)
- propriedade da escolha gulosa (*greedy-choice property*)

Problema dos intervalos disjuntos

Problema: Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$, encontrar uma coleção máxima de intervalos disjuntos dois a dois.

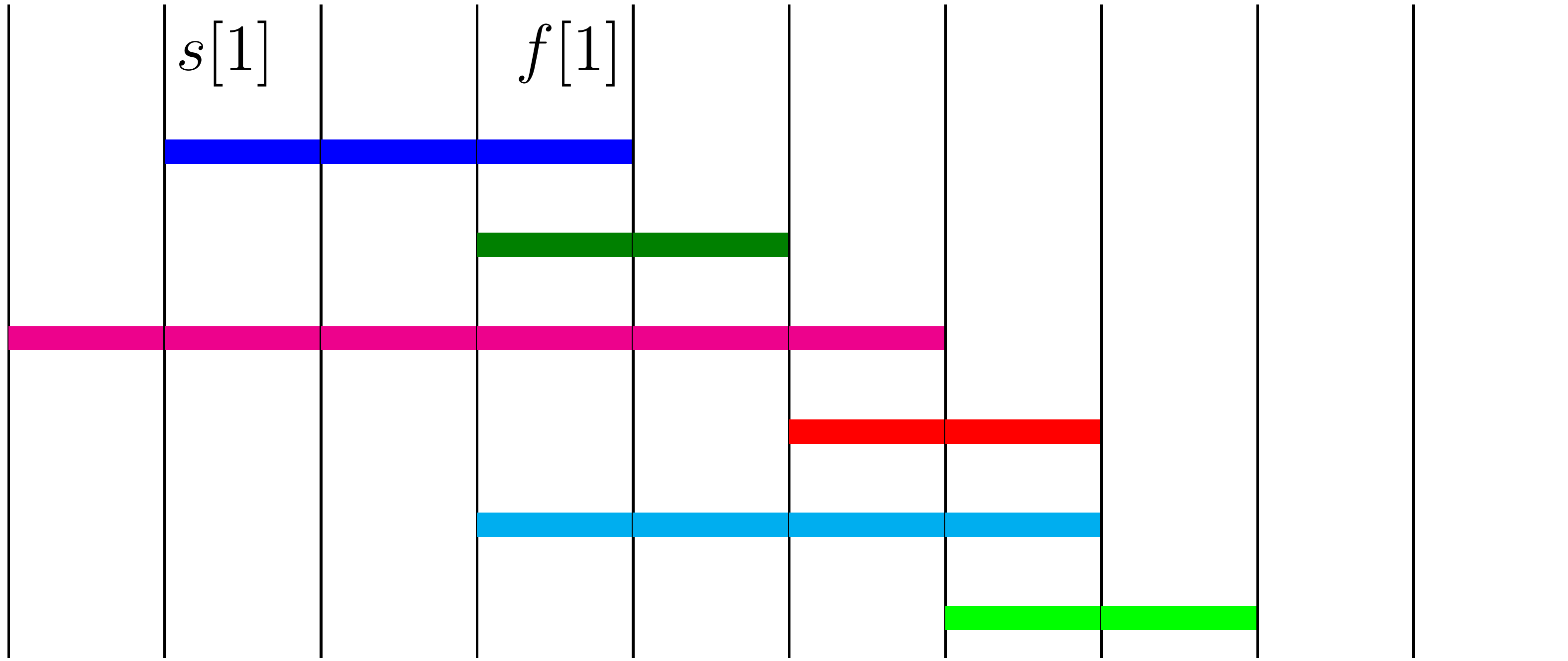
Solução é um subconjunto A de $\{1, \ldots, n\}$.

Problema dos intervalos disjuntos

Problema: Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$, encontrar uma coleção máxima de intervalos disjuntos dois a dois.

Solução é um subconjunto A de $\{1, \ldots, n\}$.

Exemplo:

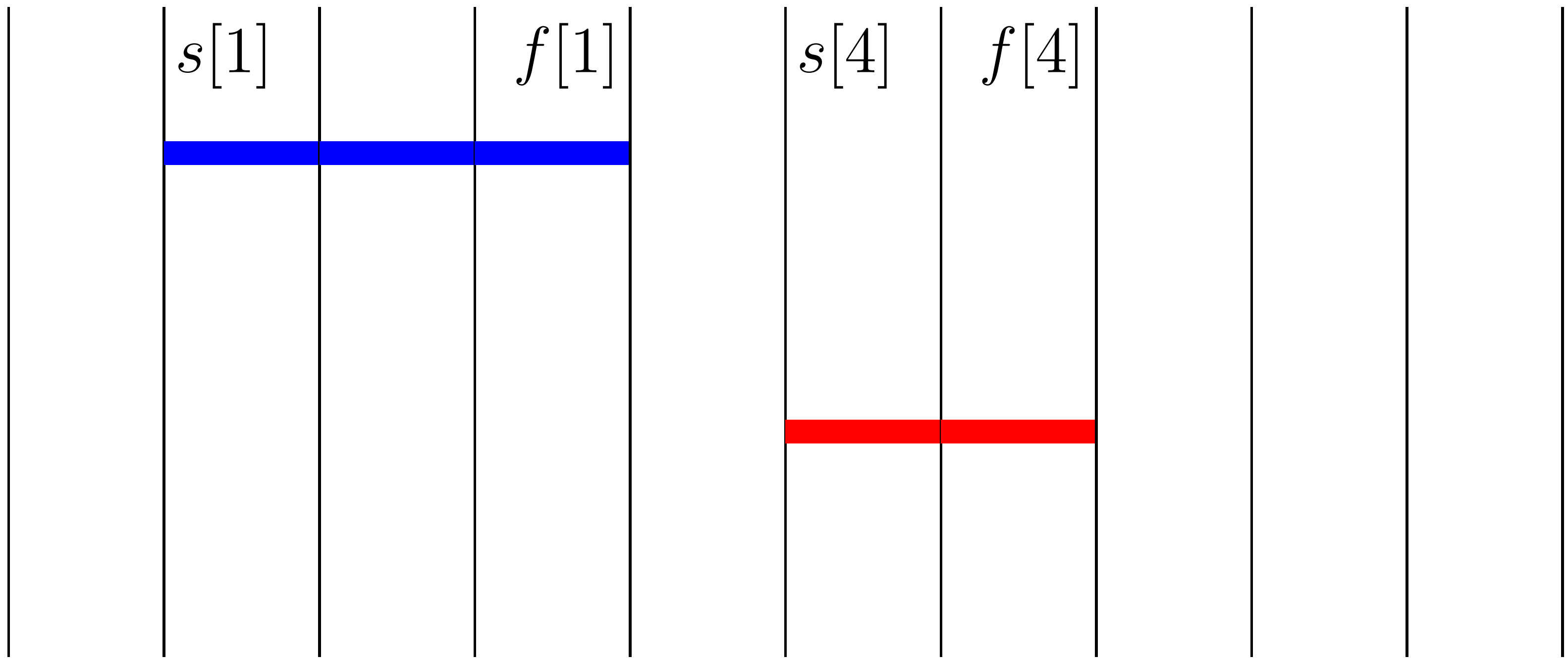


Problema dos intervalos disjuntos

Problema: Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$, encontrar uma coleção máxima de intervalos disjuntos dois a dois.

Solução é um subconjunto A de $\{1, \ldots, n\}$.

Solução:

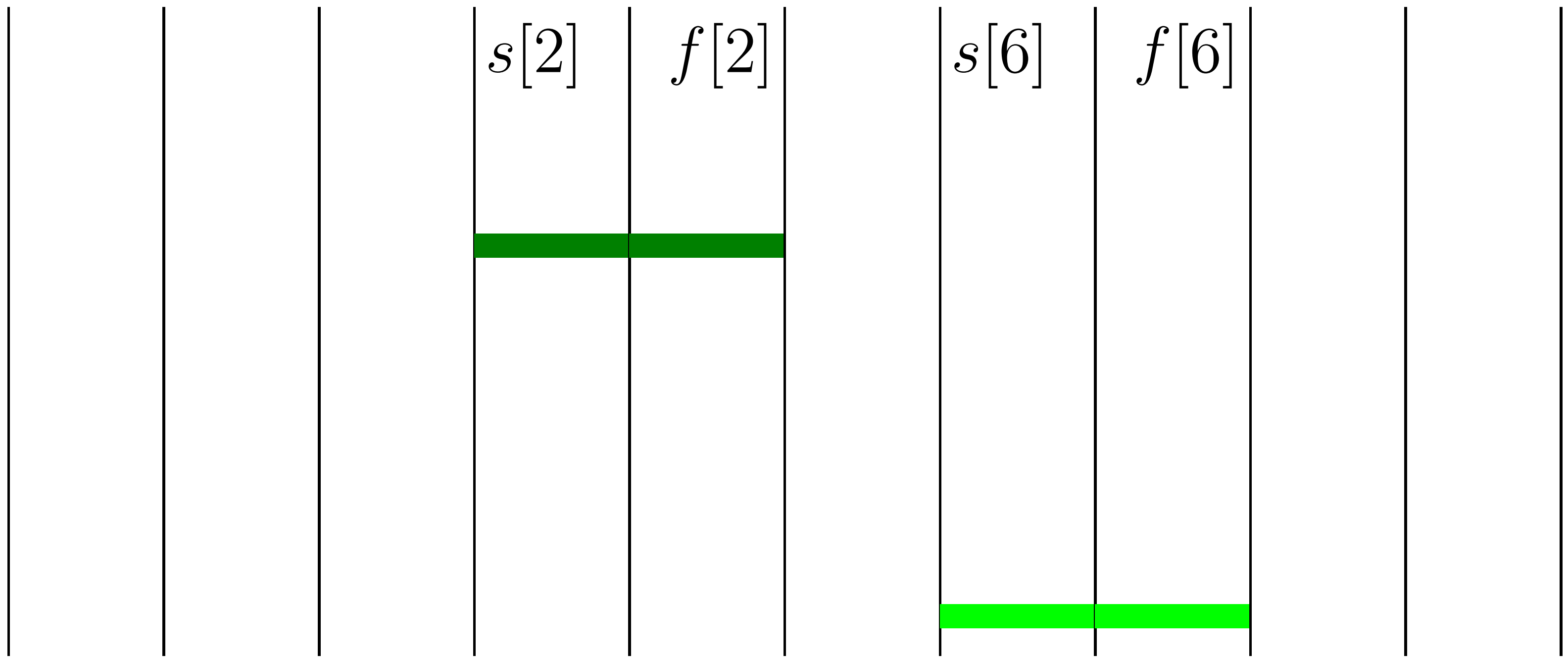


Problema dos intervalos disjuntos

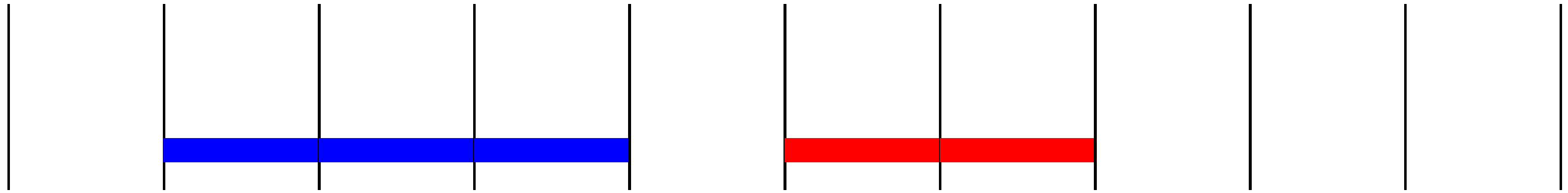
Problema: Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$, encontrar uma coleção máxima de intervalos disjuntos dois a dois.

Solução é um subconjunto A de $\{1, \ldots, n\}$.

Solução:



Motivação



Se cada intervalo é uma "atividade", queremos coleção disjunta máxima de atividades compatíveis (i e j são compatíveis se $f[i] \leq s[j]$).

Nome no CLRS: Activity Selection Problem

Algoritmo guloso

GULOSO (s, f, n)
    $A \leftarrow \emptyset$
    $R \leftarrow \{1, \ldots, n\}$
    enquanto $R \neq \emptyset$ **faça**
        escolha por um critério guloso um intervalo i de R
        $A \leftarrow A \cup \{i\}$
        $R \leftarrow R \setminus \{j \in R : j \text{ intersecta } i\}$
    devolva A

Claro que A é uma coleção de intervalos compatíveis.

Qual critério usamos para escolher um intervalo de R?

Possíveis critérios gulosos

- escolher o intervalo i em R com menor $s[i]$;

Possíveis critérios gulosos

- escolher o intervalo i em R com menor $s[i]$;

- escolher o intervalo i em R com menor $f[i] - s[i]$;

Possíveis critérios gulosos

- escolher o intervalo i em R com menor $s[i]$;
- escolher o intervalo i em R com menor $f[i] - s[i]$;
- escolher o intervalo i tal que

$$|\{j \in R : j \text{ intersecta } i\}|$$

  é o menor possível;

Possíveis critérios gulosos

- escolher o intervalo i em R com menor $s[i]$;
- escolher o intervalo i em R com menor $f[i] - s[i]$;
- escolher o intervalo i tal que

  $$|\{j \in R : j \text{ intersecta } i\}|$$

  é o menor possível;
- escolher o intervalo i em R com menor $f[i]$;

Algoritmo guloso

GULOSO (s, f, n)

ORDENE(s, f, n) $\quad \triangleright\ f[1] \leq f[2] \leq \cdots \leq f[n]$
$A \leftarrow \emptyset$
$f_A \leftarrow 0$
para $i \leftarrow 1$ **até** n **faça**
    se $s[i] \geq f_A$
        então $A \leftarrow A \cup \{i\}$
                $f_A \leftarrow f[i]$
devolva A

Algoritmo guloso

GULOSO (s, f, n)

ORDENE(s, f, n) $\triangleright$ $f[1] \leq f[2] \leq \cdots \leq f[n]$
$A \leftarrow \emptyset$
$f_A \leftarrow 0$
para $i \leftarrow 1$ **até** n **faça**
    se $s[i] \geq f_A$
        então $A \leftarrow A \cup \{i\}$
            $f_A \leftarrow f[i]$
devolva A

Consumo de tempo: $\mathrm{O}(n \lg n)$

Algoritmo guloso

GULOSO (s, f, n)

ORDENE(s, f, n) $\quad \triangleright\ f[1] \leq f[2] \leq \cdots \leq f[n]$
$A \leftarrow \emptyset$
$f_A \leftarrow 0$
para $i \leftarrow 1$ **até** n **faça**
    se $s[i] \geq f_A$
        então $A \leftarrow A \cup \{i\}$
            $f_A \leftarrow f[i]$
devolva A

Consumo de tempo: $\mathrm{O}(n \lg n)$

Por que funciona?

Algoritmo guloso

GULOSO (s, f, n)

```
ORDENE(s, f, n)   ▷ f[1] ≤ f[2] ≤ ··· ≤ f[n]
A ← ∅
f_A ← 0
para i ← 1 até n faça
    se s[i] ≥ f_A
        então A ← A ∪ {i}
              f_A ← f[i]
devolva A
```

Consumo de tempo: $\mathrm{O}(n \lg n)$

Por que funciona?

Por que A é coleção máxima de intervalos compatíveis?

Subestrutura ótima

Intervalos $S := \{1, \ldots, n\}$

Suponha que

A é coleção máxima de intervalos disjuntos de S.

Se $i \in A$

então $A \setminus \{i\}$ é coleção máxima de intervalos disjuntos de $S \setminus \{k : [s[k], f[k]) \cap [s[i], f[i]) \neq \emptyset\}$.

senão A é coleção máxima de intervalos disjuntos de $S \setminus \{i\}$.

Demonstre a propriedade.

Subestrutura ótima II

Intervalos $S := \{1, \ldots, n\}$

Suponha que

A é coleção máxima de intervalos disjunto de S.

Se $i \in A$ é tal que $f[i]$ é mínimo

então $A - \{i\}$ é coleção máxima de intervalos disjuntos de $\{k : s[k] \geq f[i]\}$.

$\{k : s[k] \geq f[i]\} =$ todos intervalos “à direita” de “i”.

Demonstre a propriedade.

Coloração de intervalos

Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$ e um inteiro positivo k, uma k-coloração dos intervalos é uma partição deles em k coleções de intervalos disjuntos.

Coloração de intervalos

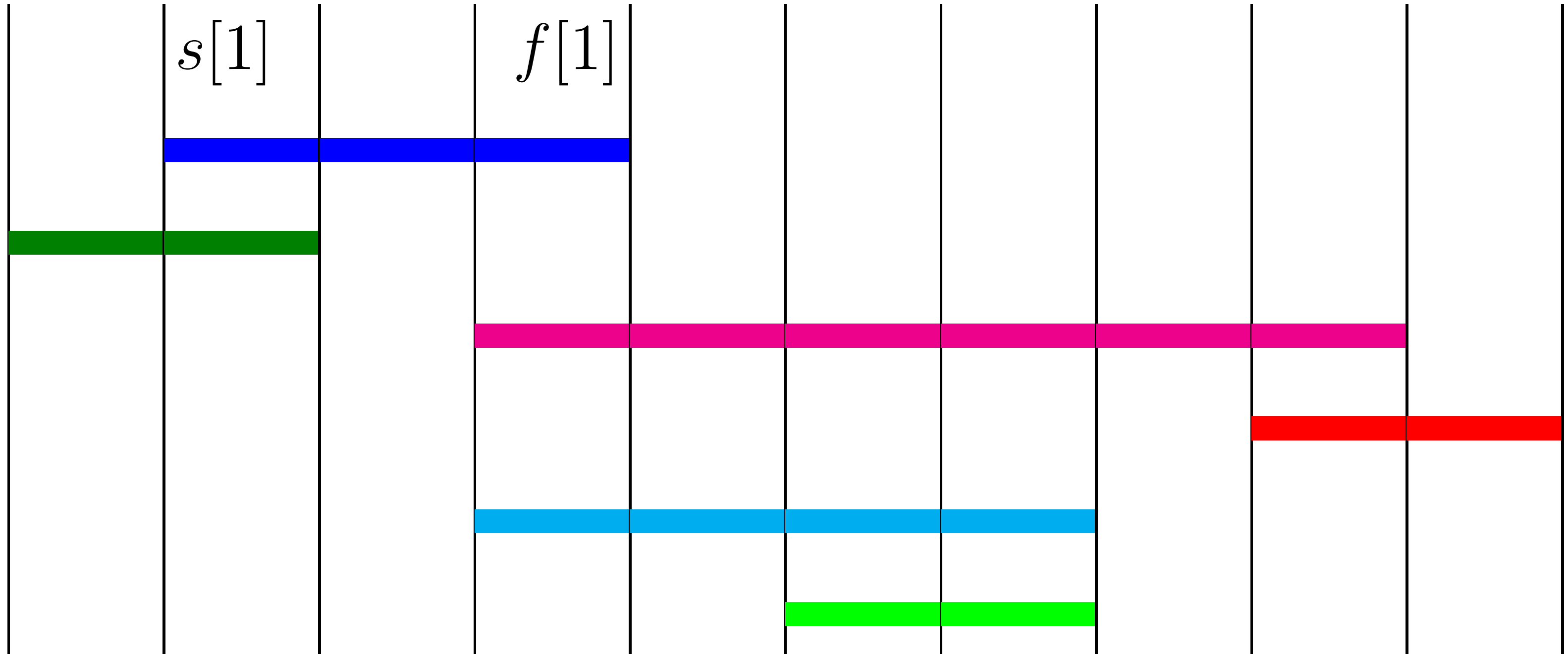
Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$ e um inteiro positivo k, uma k-coloração dos intervalos é uma partição deles em k coleções de intervalos disjuntos.

Problema: Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$, encontrar uma k-coloração dos intervalos com k o menor possível.

Coloração de intervalos

Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$ e um inteiro positivo k, uma k-coloração dos intervalos é uma partição deles em k coleções de intervalos disjuntos.

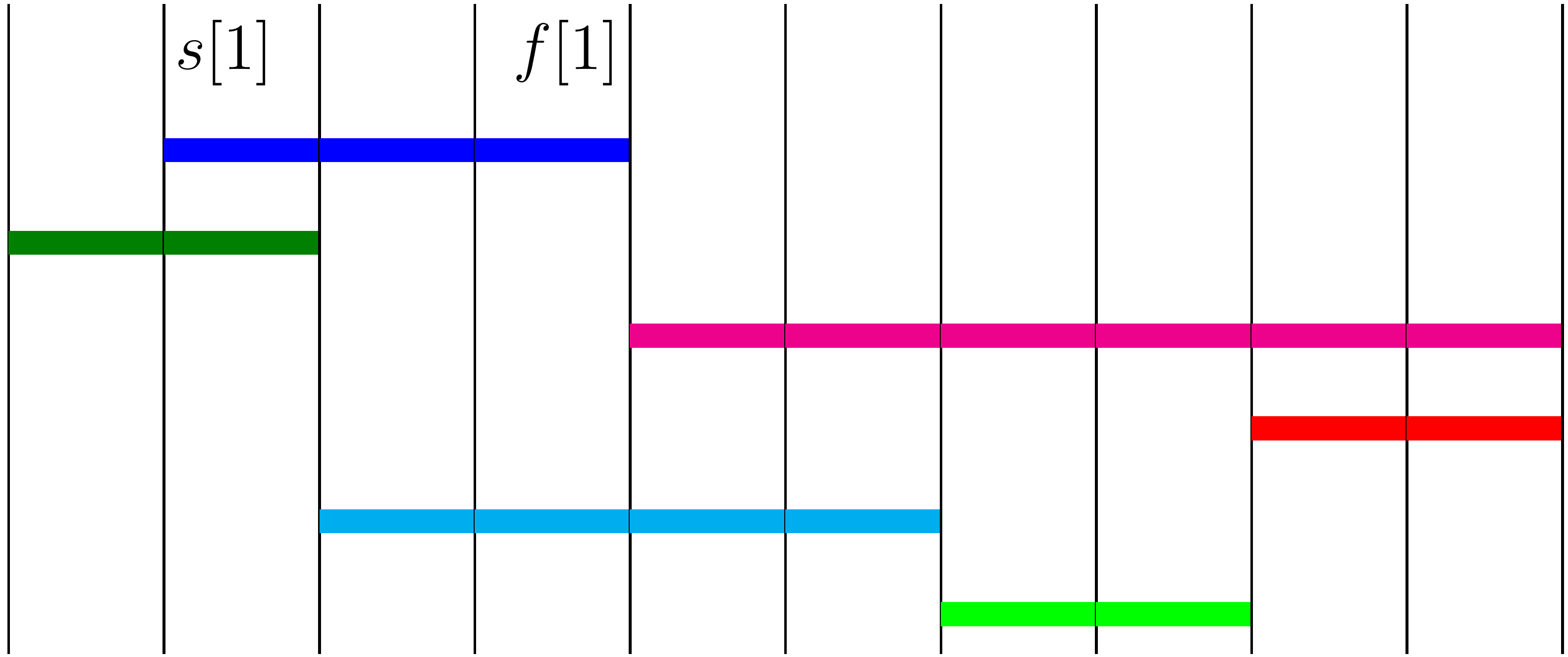
Problema: Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$, encontrar uma k-coloração dos intervalos com k o menor possível.



Coloração de intervalos

Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$ e um inteiro positivo k, uma k-coloração dos intervalos é uma partição deles em k coleções de intervalos disjuntos.

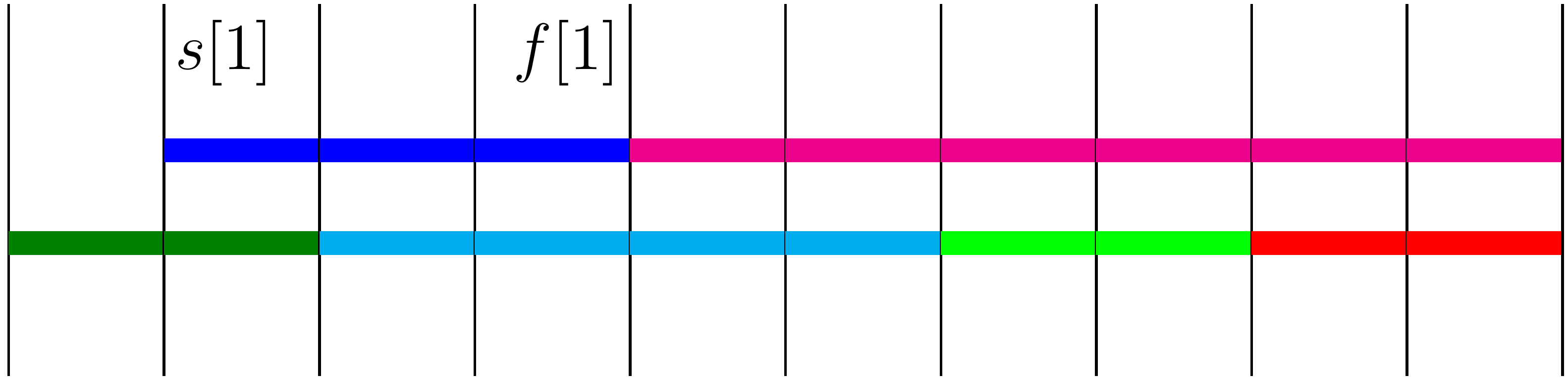
Problema: Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$, encontrar uma k-coloração dos intervalos com k o menor possível.



Coloração de intervalos

Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$ e um inteiro positivo k, uma k-coloração dos intervalos é uma partição deles em k coleções de intervalos disjuntos.

Problema: Dados intervalos $[s[1], f[1]), \ldots, [s[n], f[n])$, encontrar uma k-coloração dos intervalos com k o menor possível.



Solução: 2-coloração.

Algoritmo guloso

GULOSO (s, f, n)

ORDENE(s, f, n) $\quad \triangleright\ s[1] \le s[2] \le \cdots \le s[n]$

$m \leftarrow 0$

$\ell[1] \leftarrow 0$

para $k \leftarrow 1$ **até** n **faça**

    $j \leftarrow 1$

    enquanto $\ell[j] > s[k]$ **faça**

        $j \leftarrow j + 1$

    se $j > m$

        então $m \leftarrow j$

                $\ell[m + 1] \leftarrow 0$

    $\ell[j] \leftarrow f[k]$

    $c[k] \leftarrow j$

devolva c

Exercícios

1. Mostre um exemplo para os três primeiros critérios gulosos apresentados para o primeiro problema que prove que o algoritmo obtido usando estes critérios pode produzir um conjunto A que não é máximo.
2. Considere o algoritmo da transparência anterior para o segundo problema. Modifique-o para que, além de c, ele devolva um conjunto S de m intervalos e um instante t tal que $s[i] \leq t < f[i]$ para todo i em S.
3. Na versão apresentada em aula destas transparências, o algoritmo do segundo problema ordenava os intervalos pelo valor de f em vez de pelo valor de s. Isso faz diferença? Ou seja, a versão do algoritmo da página anterior que ordena os intervalos por f em vez de s também dá uma resposta correta? Prove ou dê um contra-exemplo.