

# Construção e Análise de Algoritmos

## 3a avaliação remota - p. síncrona

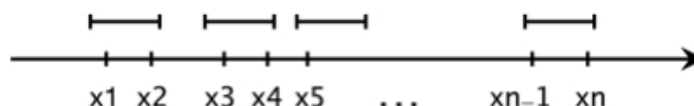
### 1. Cobertura de intervalos unitários

Considere uma sequência de pontos  $x_1, x_2, \dots, x_n$ .

Uma *cobertura de intervalos unitários* para esses pontos consiste em uma coleção de intervalos  $[a_1, b_1], [a_2, b_2], \dots, [a_n, b_n]$  tal que

- i.  $b_j - a_j = 1$ , para todo  $j$  (i.e., todos os intervalos tem tamanho 1)
- ii. todo ponto  $x_i$  pertence a algum intervalo  $[a_j, b_j]$

Abaixo nós temos uma ilustração dessa ideia



O problema consiste em encontrar a menor cobertura de intervalos unitários possível para a sequência de pontos  $x_1, x_2, \dots, x_n$ .

- a) Apresente uma estratégia gulosa para esse problema.
- b) Argumente que a sua estratégia encontra uma solução ótima para o problema.

### 2. Tempo de espera com prioridades

Imagine que  $n$  programas estão esperando para serem executados em um supercomputador. Cada programa  $p$  requer tempo de processamento  $T_p$ , e deve ser executado do início ao fim sem interrupção.

Os programas podem ser executados em qualquer ordem, mas o  $j$ -ésimo programa a ser executado terá um tempo total de espera igual a

$$E_j = \sum_{i=1}^j T_i$$

isto é, a soma do seu próprio tempo de execução com o tempo de todos os programas que são executados antes dele (esse é o tempo que o programador espera para obter o resultado do seu programa).

Até aqui, esse problema é muito semelhante àquele que aparece na lista 16.

A novidade é que agora os programas tem prioridades  $\delta_i$  associadas a eles.

E o objetivo consiste em escolher uma ordem para a execução dos programas que minimiza

a expressão

$$EP = \sum_j E_j \cdot \delta_j$$

Intuitivamente, programas com uma prioridade alta deveriam ser executados primeiro para reduzir o seu tempo de espera.

Mas, se o tempo de execução desses programas é muito grande, isso tem um impacto no tempo de espera de vários outros programas, e a ideia pode não valer a pena.

- a) Apresente uma estratégia para definir a ordem de execução dos programas que minimiza a expressão EP.
- b) Argumente que a sua estratégia encontra a melhor solução possível.

**Dica:** A estratégia gulosa ótima pode não ser evidente à primeira vista.

Mas, você pode seguir um raciocínio semelhante ao que foi feito na aula 18.

Quer dizer, comece com uma ordem qualquer e veja se você consegue reduzir o valor de EP trocando dois programas de lugar.

Considerar um exemplo concreto com apenas 3 programas também pode ajudar.

Tente entender em que condições o programa  $p_i$  deve ficar na frente do programa  $p_j$ .

### 3. Seleção de atividades circular

*A onda de hiperprodutividade finalmente chegou à universidade.*

*A administração superior divulgou há poucos dias que está trabalhando no plano UFC - 247 (i.e., funcionamento 24h por dia, 7 dias por semana).*

*Agora acabou a moleza!*

*Alguns professores se entusiasmaram com a ideia, e já estão se organizando para ministrar as suas aulas no meio da madrugada.*

*Eles imaginam que, trabalhando um pouco mais (e dormindo um pouco menos) no início do semestre, eles vão ter um pouco mais de tempo no final do semestre para poder pensar sobre alguma ideia.*

*Mas, não é claro o que a administração superior vai pensar quando ver esse pessoal sem fazer nada por aí ...*

*Seja como for, a novidade já está trazendo dores de cabeça para o coordenador.*

*Quer dizer, agora vão surgir demandas para o uso do laboratório nas 24h do dia.*

*E daí, a sua boa e velha estratégia de escolher primeiro as atividades que acabam mais cedo já não faz mais sentido.*

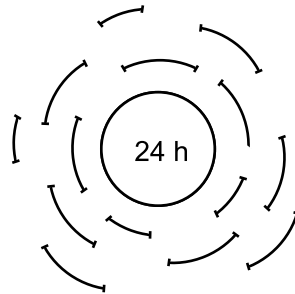
*Quer dizer, agora que os professores vão dar aulas todos os dias, de dia e de noite, o que é que vem primeiro: as 3h da tarde, as 9h da manhã, ou a meia noite?*

*De fato, como diz a administração superior, essa história de dia, noite, mais cedo e mais tarde, é tudo coisa do passado.*

*A única coisa que importa é maximizar o número de atividades.*

*E o coordenador que resolva esse problema da melhor maneira possível.*

Abaixo nós temos uma ilustração dessa nova versão do problema da seleção de atividades.



Como antes, o problema consiste em selecionar o maior número possível de atividades compatíveis (i.e., que não tenham conflito de horário).

- a) Encontre um contra-exemplo para a seguinte estratégia gulosa
- selecione a atividade com a menor duração,  
elimine as atividades incompatíveis com ela do problema,  
e resolva o problema linear resultante com o algoritmo da aula 15

Explique brevemente o seu contra-exemplo.

- b) Encontre um contra-exemplo para a seguinte estratégia gulosa
- selecione a atividade com o menor número de incompatibilidades,  
elimine as atividades incompatíveis com ela do problema,  
e resolva o problema linear resultante com o algoritmo da aula 15

Explique brevemente o seu contra-exemplo.

- c) Apresente um algoritmo eficiente (i.e., de tempo polinomial) que encontra uma solução ótima para esse problema.
- d) Explique sucintamente porque o seu algoritmo tem a garantia de encontrar a solução ótima do problema.

*Nota:* Aqui não é preciso um argumento de otimalidade completo, mas apenas uma explicação sucinta mesmo.