Algoritmos Gulosos

Análise de Algoritmos – Ciência da Computação

Prof. Daniel Saad Nogueira Nunes

IFB – Instituto Federal de Brasília, Campus Taguatinga

Sumário

1. Introdução
2. Algoritmos Gulosos
3. Framework

Sumário

1 Introdução

Introdução

Problemas de Otimização

- Em problemas de otimização, estamos procurando sempre a solução com o mínimo (máximo) valor possível.
- Nesses problemas, nos deparamos com uma série de escolhas, onde temos que escolher a adequada para chegar na melhor solução possível.

Introdução

Algoritmos Gulosos

- Um **algoritmo guloso** é aquele que olha localmente pro que se tem.
- Sempre escolhemos aquela que parece ser a melhor escolha no momento.
- Uma escolha local nem sempre resulta na solução ótima do problema, mas às vezes sim.
- Nos concentraremos em estudar os algoritmos gulosos que conseguem obter soluções ótimas para os problemas.

Sumário

2 Algoritmos Gulosos

Algoritmos Gulosos

- Vamos dar um exemplo de um algoritmo guloso que resolve o problema da Seleção de Eventos.
- Este algoritmo sempre faz a melhor escolha no momento e, mesmo assim, consegue chegar na solução ótima global.
- Antes de introduzi-lo, precisamos de algumas definições...

Algoritmos Gulosos

Definição (Evento)

- Evento: atividade disposta em um intervalo de tempo.
- Cada evento e_i, possui um tempo de início, $e_i.s$ e um tempo de fim $e_i.f$, de forma que $0 \leq e_i.s < e_i.f < \infty$.
- Dois eventos e_i e e_j são ditos **compatíveis**, se $[e_i.s, e_i.f) \cap [e_j.s, e_j.f) = \emptyset$.
- Equivalentemente, e_i e e_j são compatíveis se $e_i.s \geq e_j.f$ ou se $e_j.s \geq e_i.f$.

Algoritmos Gulosos

Problema da Seleção de Eventos

Suponha que tenhamos diversos eventos competindo por um recurso em comum.

- **Entrada:** $S = \{e_0, e_1, \ldots, e_{n-1}\}$. Um conjunto de eventos ordenados pelo tempo de término.
- Tamanho do maior conjunto de eventos que são compatíveis entre si.

Seleção de Eventos

Exemplo

Considere os seguintes eventos:

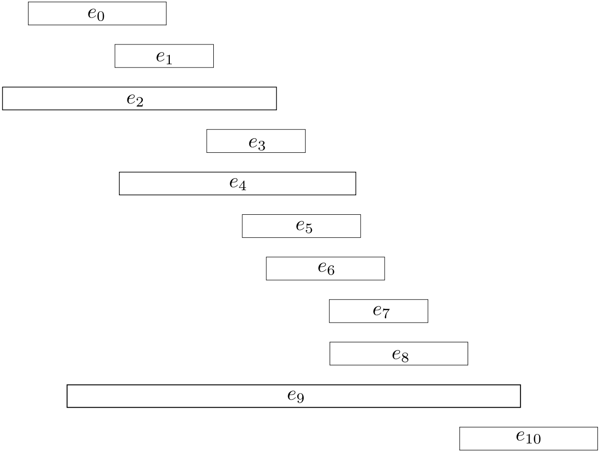
Tabela: Eventos.

i	0	1	2	3	4	5	6	7	8	9	10
$e[i].s$	1	3	0	5	3	5	6	8	8	2	12
$e[i].f$	4	5	6	7	9	9	10	11	12	14	16

- Qual é o maior tamanho possível de conjunto compatível de atividades?

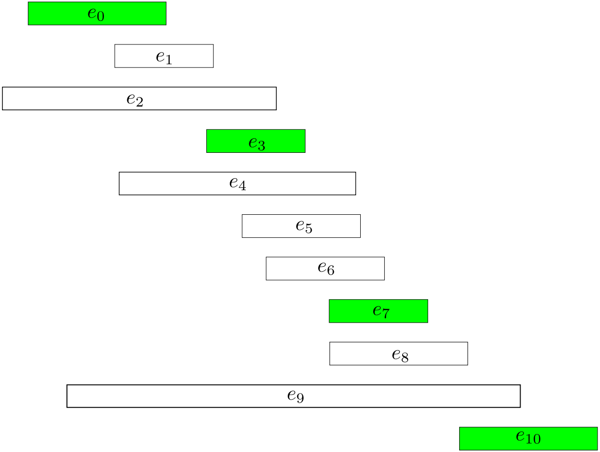
Seleção de Eventos

Exemplo



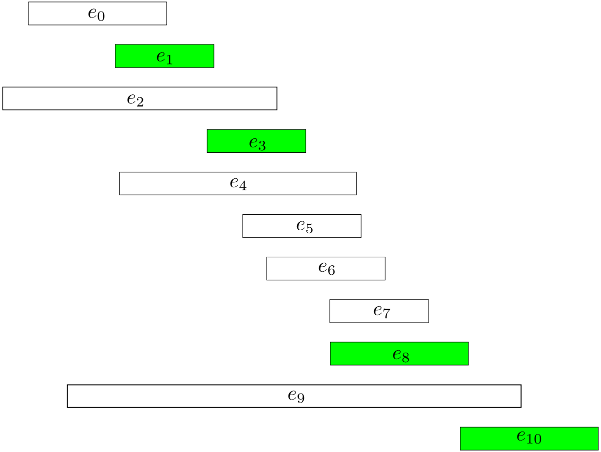
Seleção de Eventos

Exemplo



Seleção de Eventos

Exemplo



Subestrutura ótima

- Vamos verificar que o problema da Seleção de Eventos tem uma **subestrutura ótima**.
- Seja $S_{i,j}$ o conjunto de eventos que começa depois que o evento e_i termina e que termina antes do evento e_j começar.
- Queremos encontrar o conjunto maximal de eventos compatíveis em $S_{i,j}$. Chamaremos esse conjunto de $A_{i,j}$.
  - Suponha que $e_k \in A_{i,j}$.

Subestrutura ótima

- Como e_k está na solução ótima. Temos que resolver dois problemas.
  - Descobrir o maior conjunto compatível de $S_{i,k}$.
  - Descobrir o maior conjunto compatível de $S_{k,j}$
- Seja $A_{i,k} = A_{i,j} \cap S_{i,k}$.
- Seja $A_{k,j} = A_{i,j} \cap S_{k,j}$.
- $A_{i,k}$: o conjunto de eventos em $A_{i,j}$ que começam após e_i e terminam antes de e_k começar.
- $A_{k,j}$ tem o conjunto de eventos em $A_{i,j}$ que começam após e_k e terminam antes de e_j começar.
- $\therefore A_{i,j} = A_{i,k} \cup \{e_k\} \cup A_{k,j}$

Subestrutura ótima

- Podemos concluir disso que a solução ótima $A_{i,j}$ tem tamanho $|A_{i,k}| + 1 + |A_{k,j}|$.
- Tanto $|A_{i,k}|$ quando $|A_{k,j}|$ devem ser soluções ótimas, caso contrário, conseguiríamos obter um $|A_{i,j}|$ maior.

Solução

- Podemos resolver o problema recursivamente com base na seguinte relação de recorrência:

$$T(i,j) = \begin{cases} 0, & \text{se } S_{i,j} = \emptyset \\ \max\limits_{a_k \in S_{i,j}} \{T(i,k) + 1 + T(k,j)\} \end{cases}$$

- A solução funciona, mas estaremos ignorando totalmente a natureza do problema. . .

Escolha Gulosa

- O que o problema intuitivamente nos diz?

Escolha Gulosa

- O que o problema intuitivamente nos diz?
- Que se escolhermos um evento que deixa o máximo de recursos para os outros, conseguiremos a solução ótima.

Escolha Gulosa

Teorema

Considere um problema não vazio S_k e seja e_m um evento em S_k com o tempo de término mais baixo. e_m tem que estar em um conjunto máximo de eventos compatíveis de S_k.

Escolha Gulosa

Demonstração

Seja A_k um conjunto maximal de eventos compatíveis em S_k. Tome e_j como a atividade em A_k com menor tempo de término. Se $e_j = e_m$, finalizamos a prova. Se $e_j \neq e_m$, tome o conjunto $A'_k = A_k - \{e_j\} \cup \{e_m\}$ (estamos substituindo e_j por e_m). Os eventos em A'_k são compatíveis, uma vez que e_j foi trocado por e_m e $e_m.f \leq e_j.f$. Concluímos então que $|A'_k| = |A_k|$, e portanto A'_k também tem que ser uma solução ótima.

□

Escolha Gulosa

- O que podemos concluir disso?
- O elemento com menor tempo de término está em uma solução ótima.
- Pegamos o problema com menor tempo de término, incluímos na solução, e resolvemos um subproblema menor usando a mesma estratégia de modo que a solução do subproblema seja compatível com o elemento retirado.
- Escolha gulosa! Estamos sempre retirando um cara com uma certa propriedade.

Seleção de Eventos

Algorithm 1: RECURSIVE-GREEDY-EVENT-SELECTOR

Input: $e[1, n], k$

Output: A, conjunto maximal de eventos compatíveis

$m \leftarrow k + 1$

while $(m < n) \wedge (e[m].s < e[k].f)$ **do**

  $m++$

if($m < n$)

  return $e_m \cup$ RECURSIVE-GREEDY-EVENT-SELECTOR(e, m)

return $\emptyset$

Chamada inicial: RECURSIVE-EVENT-SELECTOR$(e, 0)$ Observação: e_0 é um elemento artificial com $e[0].f = 0$.

Seleção de Eventos

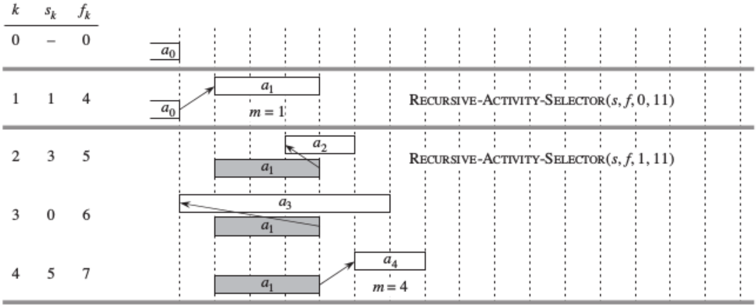


Figura: Seleção de Eventos.

Seleção de Eventos

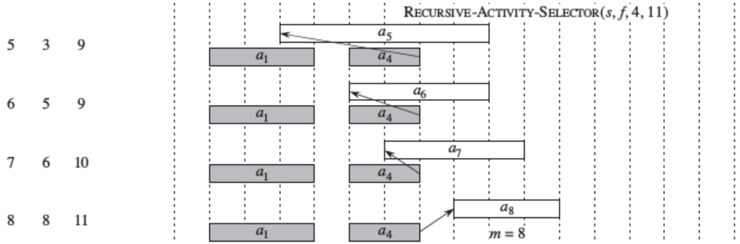


Figura: Seleção de Eventos.

Seleção de Eventos

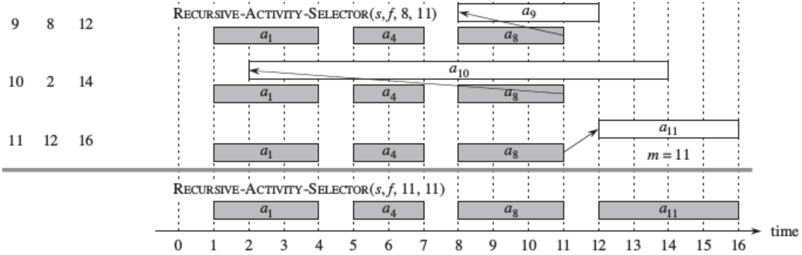


Figura: Seleção de Eventos.

Seleção de Eventos

- Qual a complexidade do algoritmo?
- Para responder essa pergunta, basta analisar quantas vezes cada evento é checado.

Seleção de Eventos

- Qual a complexidade do algoritmo?
- Para responder essa pergunta, basta analisar quantas vezes cada evento é checado.
- Cada evento é checado 1 vez. Complexidade $\Theta(n)$.

Seleção de Eventos

- Apesar de poder ser implementado recursivamente, implementaremos iterativamente usando a mesma ideia.

Seleção de Eventos

Algorithm 2: GREEDY-EVENT-SELECTOR

Input: $e[0, n-1]$

Output: A, conjunto maximal de eventos compatíveis

$A \leftarrow \{e_0\}$
$k \leftarrow 0$
for($i \leftarrow 1; i < n; i++$)
    if($e[i].s \geq e[k].f$)
        $A \leftarrow A \cup \{e_i\}$
        $k \leftarrow i$
return A

Sumário

3 Framework

Framework de Construção de Algoritmos Gulosos

1. Modele o problema de modo que seja feita uma escolha e sobre um subproblema para resolver.
2. Mostre que existe sempre uma solução ótima para o problema que admite uma escolha gulosa.
3. Demonstre que o problema tem a propriedade de subestrutura ótima ao mostrar que, ao fazer a escolha gulosa, o subproblema restante possui a propriedade que, sua solução ótima com a escolha gulosa feita anteriormente, gera uma solução ótima do problema original.