Aula 13 – Algoritmos: Algoritmos Gulosos

Norton T. Roman & Luciano A. Digiampietri
digiampietri@usp.br
@digiampietri

2023

Algoritmos Gulosos

São algoritmos que, a cada decisão:

Algoritmos Gulosos

São algoritmos que, a cada decisão:

- Sempre escolhem a alternativa que **parece** mais promissora naquele instante

Algoritmos Gulosos

São algoritmos que, a cada decisão:

- Sempre escolhem a alternativa que **parece** mais promissora naquele instante
- **Nunca** reconsideram essa decisão

Algoritmos Gulosos

São algoritmos que, a cada decisão:

- Sempre escolhem a alternativa que **parece** mais promissora naquele instante
- **Nunca** reconsideram essa decisão
  - Uma escolha que foi feita nunca é revista

Algoritmos Gulosos

São algoritmos que, a cada decisão:

- Sempre escolhem a alternativa que **parece** mais promissora naquele instante
- **Nunca** reconsideram essa decisão
  - Uma escolha que foi feita nunca é revista
  - Não há backtracking

Algoritmos Gulosos

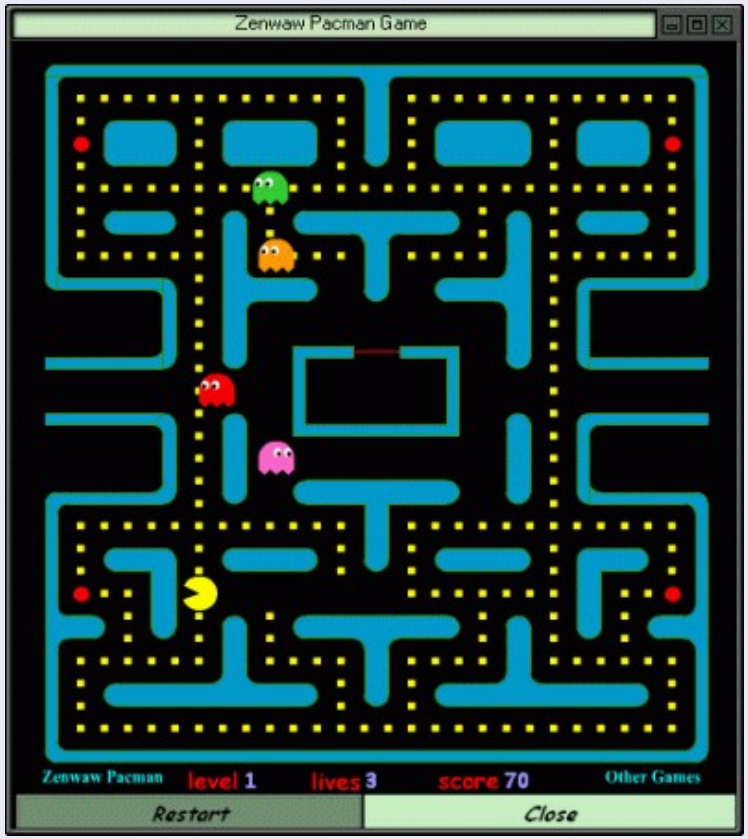
São algoritmos que, a cada decisão:

- Sempre escolhem a alternativa que **parece** mais promissora naquele instante
- **Nunca** reconsideram essa decisão
  - Uma escolha que foi feita nunca é revista
  - Não há backtracking

Algoritmos Gulosos

São algoritmos que, a cada decisão:

- Sempre escolhem a alternativa que **parece** mais promissora naquele instante
- **Nunca** reconsideram essa decisão
  - Uma escolha que foi feita nunca é revista
  - Não há backtracking



Algoritmos Gulosos

Alternativa mais promissora?

Algoritmos Gulosos

Alternativa mais promissora?

- Depende do problema, do que se quer maximizar ou minimizar

Algoritmos Gulosos

Alternativa mais promissora?

- Depende do problema, do que se quer maximizar ou minimizar
  - Ex: caminho mais curto, menor número de jogadas etc

Algoritmos Gulosos

Alternativa mais promissora?

- Depende do problema, do que se quer maximizar ou minimizar
  - Ex: caminho mais curto, menor número de jogadas etc
- Há que se ter um modo de avaliar as diferentes opções

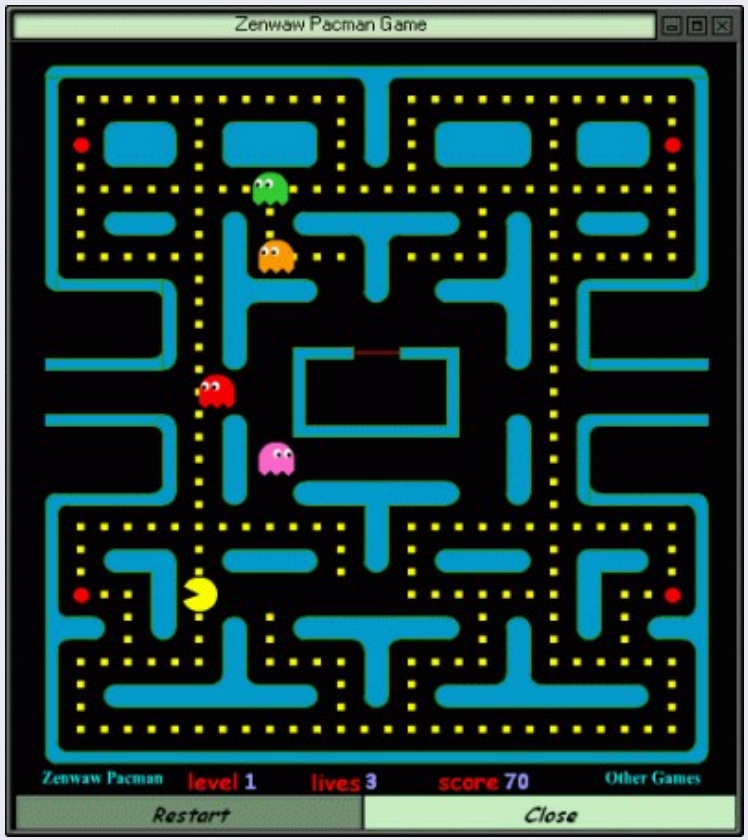
Algoritmos Gulosos

Alternativa mais promissora?

- Depende do problema, do que se quer maximizar ou minimizar
  - Ex: caminho mais curto, menor número de jogadas etc
- Há que se ter um modo de avaliar as diferentes opções
  - Uma função que diga qual delas vale mais, diante do que se considera importante para o problema

Algoritmos Gulosos

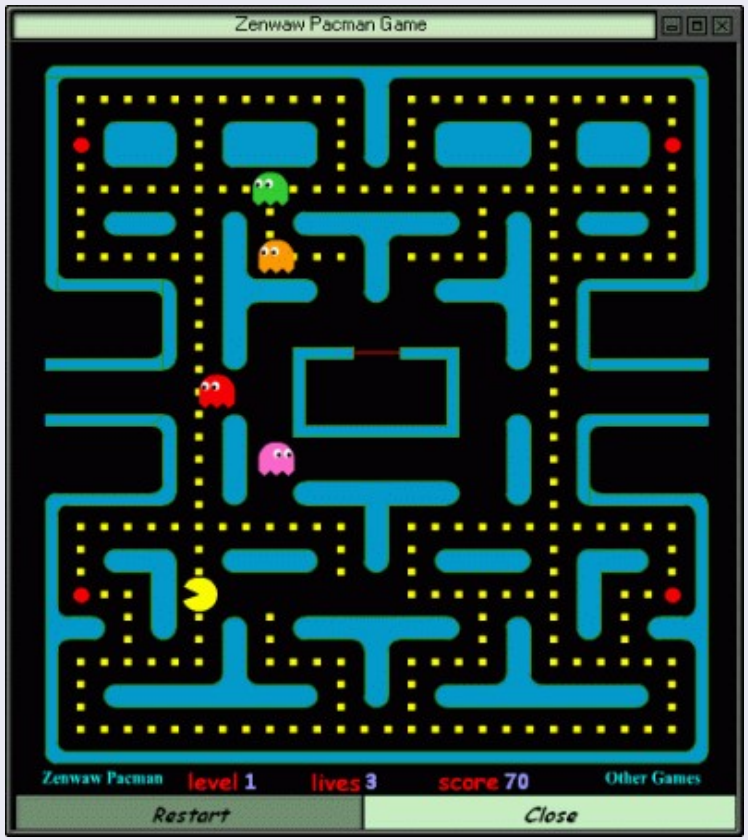
E por que "guloso"?



Algoritmos Gulosos

E por que "guloso"?

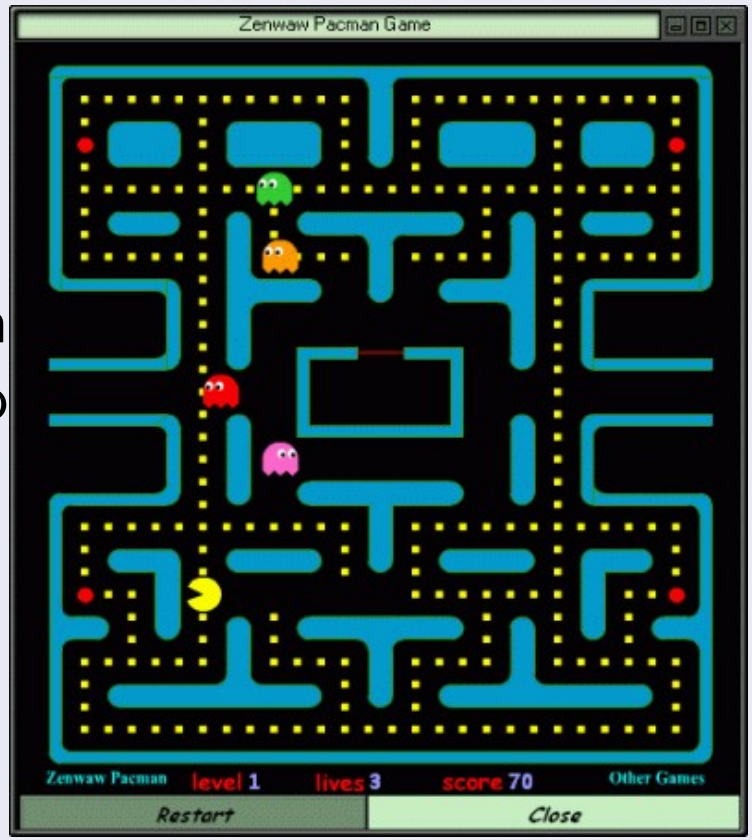
- Porque o algoritmo faz, a cada passo, a escolha que parece ser a melhor



Algoritmos Gulosos

E por que "guloso"?

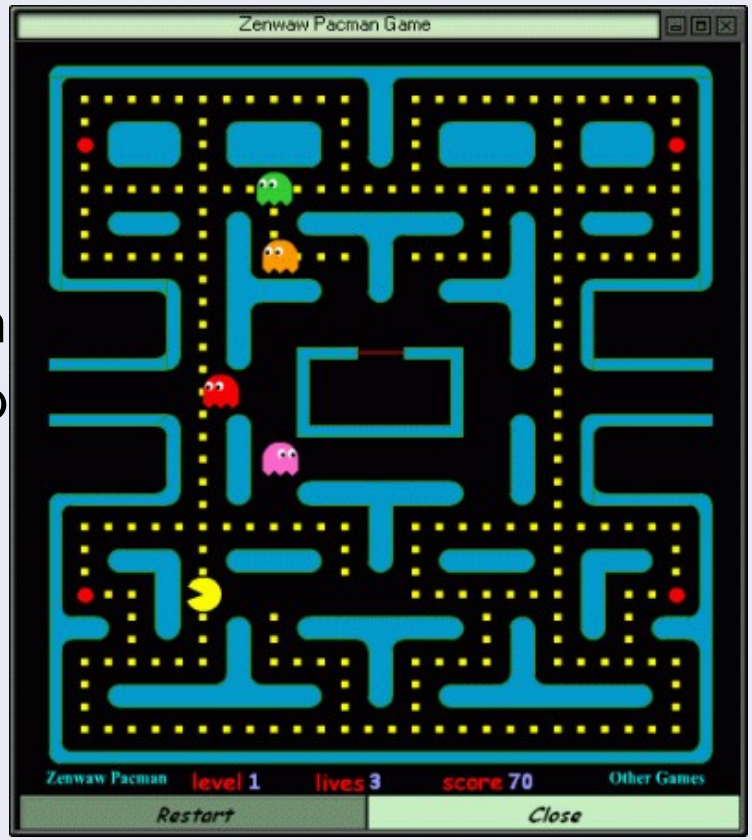
- Porque o algoritmo faz, a cada passo, a escolha que parece ser a melhor
  - Diz-se então que a escolha é feita de acordo com um critério guloso



Algoritmos Gulosos

E por que "guloso"?

- Porque o algoritmo faz, a cada passo, a escolha que parece ser a melhor
  - Diz-se então que a escolha é feita de acordo com um critério guloso
- Ou seja, faz uma escolha localmente ótima, na esperança que ela leve à solução globalmente ótima



Algoritmos Gulosos

Características

Algoritmos Gulosos

Características

- Para construir a solução ótima existe um conjunto ou lista de candidatos (passos para a solução)

Algoritmos Gulosos

Características

- Para construir a solução ótima existe um conjunto ou lista de candidatos (passos para a solução)
- Na medida em que o algoritmo procede, são acumulados um conjunto de candidatos considerados e escolhidos, e o outro de candidatos considerados e rejeitados

Algoritmos Gulosos

Características

- Para construir a solução ótima existe um conjunto ou lista de candidatos (passos para a solução)
- Na medida em que o algoritmo procede, são acumulados um conjunto de candidatos considerados e escolhidos, e o outro de candidatos considerados e rejeitados
- Existe uma função que verifica se um conjunto particular de candidatos produz uma solução para o problema (sem considerar otimalidade no momento)

Algoritmos Gulosos

Características

- Outra função verifica se um conjunto de candidatos é viável (também sem preocupar com a otimalidade)

Algoritmos Gulosos

Características

- Outra função verifica se um conjunto de candidatos é viável (também sem preocupar com a otimalidade)
  - Se é possível chegar a uma solução com o candidato

Algoritmos Gulosos

Características

- Outra função verifica se um conjunto de candidatos é viável (também sem preocupar com a otimalidade)
  - Se é possível chegar a uma solução com o candidato
- Uma função de seleção indica a qualquer momento quais dos candidatos restantes é o mais promissor

Algoritmos Gulosos

Características

- Outra função verifica se um conjunto de candidatos é viável (também sem preocupar com a otimalidade)
  - Se é possível chegar a uma solução com o candidato
- Uma função de seleção indica a qualquer momento quais dos candidatos restantes é o mais promissor
- Uma função objetivo fornece o valor da solução encontrada

Algoritmos Gulosos

Características

- Outra função verifica se um conjunto de candidatos é viável (também sem preocupar com a otimalidade)
  - Se é possível chegar a uma solução com o candidato
- Uma função de seleção indica a qualquer momento quais dos candidatos restantes é o mais promissor
- Uma função objetivo fornece o valor da solução encontrada
  - Como o comprimento do caminho construído

Algoritmos Gulosos

Características

- Outra função verifica se um conjunto de candidatos é viável (também sem preocupar com a otimalidade)
  - Se é possível chegar a uma solução com o candidato
- Uma função de seleção indica a qualquer momento quais dos candidatos restantes é o mais promissor
- Uma função objetivo fornece o valor da solução encontrada
  - Como o comprimento do caminho construído
  - Não aparece de forma explícita no algoritmo guloso

Algoritmos Gulosos

Observações

- A função de seleção é geralmente relacionada com a função objetivo

Algoritmos Gulosos

Observações

- A função de seleção é geralmente relacionada com a função objetivo
- Se o objetivo é:

Algoritmos Gulosos

Observações

- A função de seleção é geralmente relacionada com a função objetivo
- Se o objetivo é:
  - maximizar $\Rightarrow$ escolherá o candidato restante que proporcione o maior ganho individual

Algoritmos Gulosos

Observações

- A função de seleção é geralmente relacionada com a função objetivo
- Se o objetivo é:
  - maximizar $\Rightarrow$ escolherá o candidato restante que proporcione o maior ganho individual
  - minimizar $\Rightarrow$ então será escolhido o candidato restante de menor custo

Algoritmos Gulosos

Observações

- O algoritmo nunca muda de ideia

Algoritmos Gulosos

Observações

- O algoritmo nunca muda de ideia
  - Uma vez que um candidato é escolhido e adicionado à solução, ele lá permanece para sempre

Algoritmos Gulosos

Observações

- O algoritmo nunca muda de ideia
  - Uma vez que um candidato é escolhido e adicionado à solução, ele lá permanece para sempre
  - Uma vez que um candidato é excluído do conjunto solução, ele nunca mais é reconsiderado

Algoritmos Gulosos

Observações

- O algoritmo nunca muda de ideia
  - Uma vez que um candidato é escolhido e adicionado à solução, ele lá permanece para sempre
  - Uma vez que um candidato é excluído do conjunto solução, ele nunca mais é reconsiderado
- Em geral, um dos "segredos" dos algoritmos gulosos é a escolha de como o conjunto de entrada será ordenado

Algoritmos Gulosos

Observações

- Tipicamente, algoritmos gulosos são utilizados para resolver problemas de otimização que funcionem através de uma sequência de passos

Algoritmos Gulosos

Observações

- Tipicamente, algoritmos gulosos são utilizados para resolver problemas de otimização que funcionem através de uma sequência de passos
- Nem sempre dão soluções ótimas, embora muitas vezes o façam

Algoritmos Gulosos

Observações

- Tipicamente, algoritmos gulosos são utilizados para resolver problemas de otimização que funcionem através de uma sequência de passos
- Nem sempre dão soluções ótimas, embora muitas vezes o façam

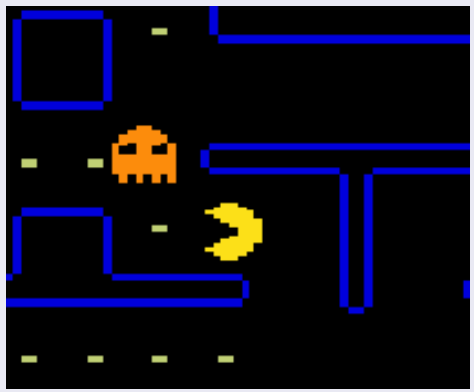
Para onde ir? 

Algoritmos Gulosos

Observações

- Tipicamente, algoritmos gulosos são utilizados para resolver problemas de otimização que funcionem através de uma sequência de passos
- Nem sempre dão soluções ótimas, embora muitas vezes o façam

Ficaria mais fácil decidir se tivéssemos mais informação

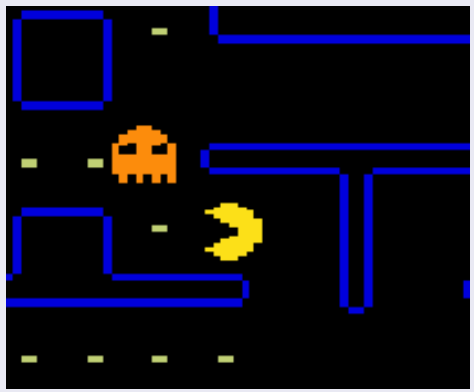


Algoritmos Gulosos

Observações

- Tipicamente, algoritmos gulosos são utilizados para resolver problemas de otimização que funcionem através de uma sequência de passos
- Nem sempre dão soluções ótimas, embora muitas vezes o façam
  - Se pudermos provar que a escolha gulosa, combinada com as escolhas feitas até então, é ótima, então ele dará a resposta ótima

Ficaria mais fácil decidir se tivéssemos mais informação



Algoritmos Gulosos

- A ideia básica da estratégia gulosa é construir, por etapas, uma solução ótima

Algoritmos Gulosos

- A ideia básica da estratégia gulosa é construir, por etapas, uma solução ótima
- Em cada passo, após selecionar o melhor elemento da entrada, decide-se se ele é viável ou não

Algoritmos Gulosos

- A ideia básica da estratégia gulosa é construir, por etapas, uma solução ótima
- Em cada passo, após selecionar o melhor elemento da entrada, decide-se se ele é viável ou não
  - Se viável, o elemento fará parte da solução

Algoritmos Gulosos

- A ideia básica da estratégia gulosa é construir, por etapas, uma solução ótima
- Em cada passo, após selecionar o melhor elemento da entrada, decide-se se ele é viável ou não
  - Se viável, o elemento fará parte da solução
- Após uma sequência de decisões, uma solução para o problema é alcançada

Algoritmos Gulosos

- A ideia básica da estratégia gulosa é construir, por etapas, uma solução ótima
- Em cada passo, após selecionar o melhor elemento da entrada, decide-se se ele é viável ou não
  - Se viável, o elemento fará parte da solução
- Após uma sequência de decisões, uma solução para o problema é alcançada
- Nessa sequência de decisões, nenhum elemento é examinado mais de uma vez

Algoritmos Gulosos

- A ideia básica da estratégia gulosa é construir, por etapas, uma solução ótima
- Em cada passo, após selecionar o melhor elemento da entrada, decide-se se ele é viável ou não
  - Se viável, o elemento fará parte da solução
- Após uma sequência de decisões, uma solução para o problema é alcançada
- Nessa sequência de decisões, nenhum elemento é examinado mais de uma vez
  - Ou ele fará parte da solução, ou será descartado

Algoritmos Gulosos

Ingredientes-chave do problema:

Algoritmos Gulosos

Ingredientes-chave do problema:

- Característica gulosa:

Algoritmos Gulosos

Ingredientes-chave do problema:

- Característica gulosa:
  - A solução ótima global pode ser produzida a partir de uma escolha ótima local

Algoritmos Gulosos

Ingredientes-chave do problema:

- Característica gulosa:
  - A solução ótima global pode ser produzida a partir de uma escolha ótima local
  - Ou seja, quando precisamos fazer uma escolha, podemos fazer aquela que parece a melhor naquele momento (a escolha da melhor opção dentre as existentes)

Algoritmos Gulosos

Ingredientes-chave do problema:

- Característica gulosa:
  - A solução ótima global pode ser produzida a partir de uma escolha ótima local
  - Ou seja, quando precisamos fazer uma escolha, podemos fazer aquela que parece a melhor naquele momento (a escolha da melhor opção dentre as existentes)
- Subestrutura ótima:

Algoritmos Gulosos

Ingredientes-chave do problema:

- Característica gulosa:
  - A solução ótima global pode ser produzida a partir de uma escolha ótima local
  - Ou seja, quando precisamos fazer uma escolha, podemos fazer aquela que parece a melhor naquele momento (a escolha da melhor opção dentre as existentes)
- Subestrutura ótima:
  - Um problema exibe subestrutura ótima se uma solução ótima para o problema contém, dentro dela, soluções ótimas para seus subproblemas

Algoritmos Gulosos

Ingredientes-chave do problema:

- Se pudermos demonstrar que o problema tem essas propriedades, então podemos construir um algoritmo guloso ótimo para ele

Algoritmos Gulosos

Ingredientes-chave do problema:

- Se pudermos demonstrar que o problema tem essas propriedades, então podemos construir um algoritmo guloso ótimo para ele
  - Precisamos demonstrar que uma solução ótima para os subproblemas, combinada com a escolha gulosa atual, leva a uma solução ótima para o problema

Algoritmos Gulosos

Ingredientes-chave do problema:

- Se pudermos demonstrar que o problema tem essas propriedades, então podemos construir um algoritmo guloso ótimo para ele
  - Precisamos demonstrar que uma solução ótima para os subproblemas, combinada com a escolha gulosa atual, leva a uma solução ótima para o problema
  - Usamos então indução nos subproblemas para mostrar que fazer a escolha gulosa em cada passo produz uma solução ótima

Algoritmos Gulosos

Ingredientes-chave do problema:

- Se pudermos demonstrar que o problema tem essas propriedades, então podemos construir um algoritmo guloso ótimo para ele
  - Precisamos demonstrar que uma solução ótima para os subproblemas, combinada com a escolha gulosa atual, leva a uma solução ótima para o problema
  - Usamos então indução nos subproblemas para mostrar que fazer a escolha gulosa em cada passo produz uma solução ótima
- Do contrário, ainda podemos chegar a uma solução, embora não ótima

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- Existem diversas atividades (por exemplo, aulas) que querem usar um mesmo recurso (por exemplo, uma sala de aulas)

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- Existem diversas atividades (por exemplo, aulas) que querem usar um mesmo recurso (por exemplo, uma sala de aulas)
- Cada atividade tem um horário de início e um horário de fim

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- Existem diversas atividades (por exemplo, aulas) que querem usar um mesmo recurso (por exemplo, uma sala de aulas)
- Cada atividade tem um horário de início e um horário de fim
- Só existe uma sala disponível

Algoritmos Gulosos – Exemplos

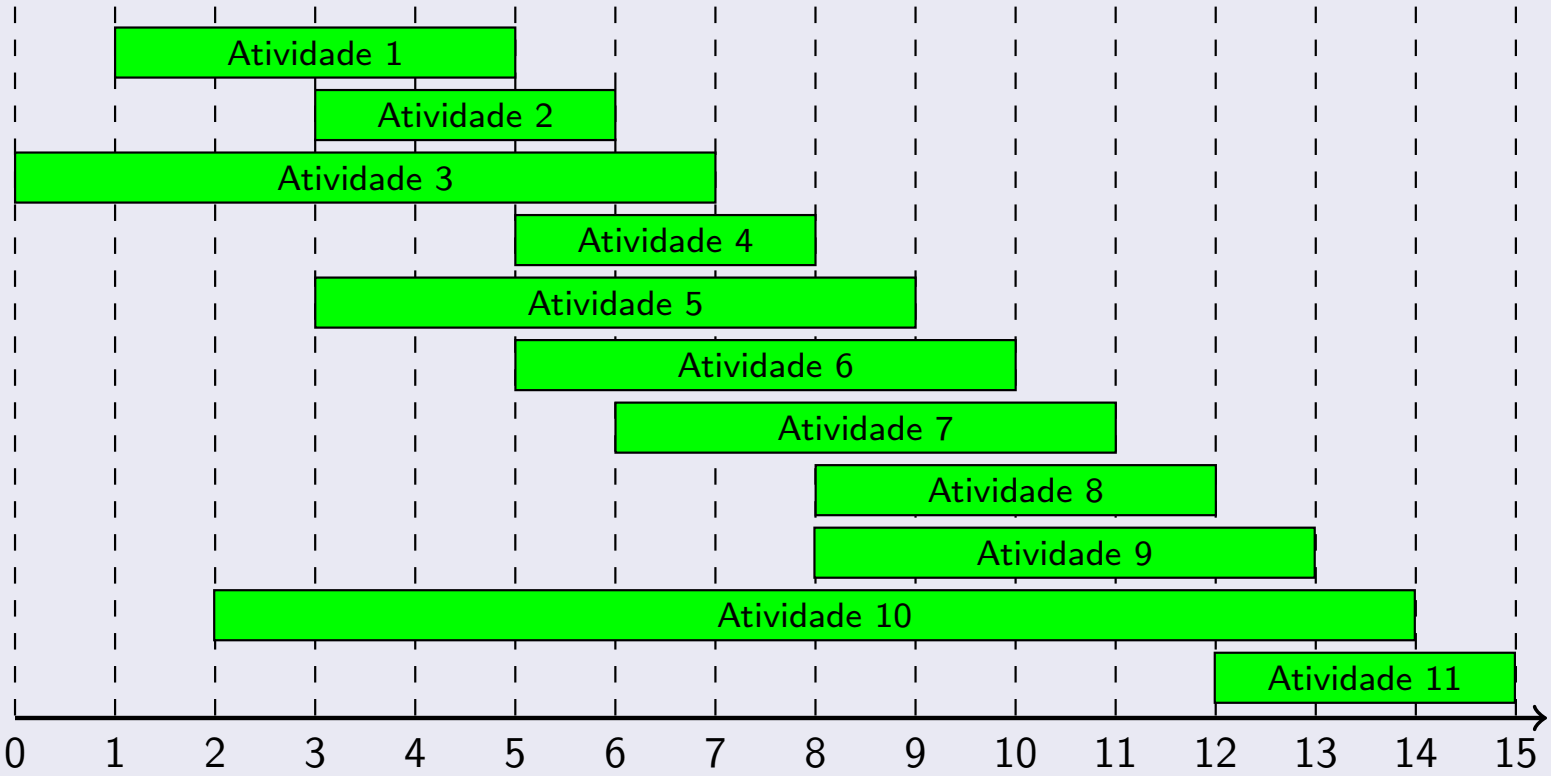
Exemplo: Seleção de Atividades

- Existem diversas atividades (por exemplo, aulas) que querem usar um mesmo recurso (por exemplo, uma sala de aulas)
- Cada atividade tem um horário de início e um horário de fim
- Só existe uma sala disponível
- Duas aulas não podem podem ser ministradas na mesma sala ao mesmo tempo

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- Ex: 11 atividades, em 15 unidades de tempo



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- Objetivo: selecionar um conjunto máximo de atividades compatíveis

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- Objetivo: selecionar um conjunto máximo de atividades compatíveis
  - Atividades compatíveis são atividades que não têm sobreposição de tempo

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- Objetivo: selecionar um conjunto máximo de atividades compatíveis
  - Atividades compatíveis são atividades que não têm sobreposição de tempo
- Ou seja, criar o maior grupo de atividades sem que haja sobreposição de tempo

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- Objetivo: selecionar um conjunto máximo de atividades compatíveis
  - Atividades compatíveis são atividades que não têm sobreposição de tempo
- Ou seja, criar o maior grupo de atividades sem que haja sobreposição de tempo
- Como fazer?

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- Objetivo: selecionar um conjunto máximo de atividades compatíveis
  - Atividades compatíveis são atividades que não têm sobreposição de tempo
- Ou seja, criar o maior grupo de atividades sem que haja sobreposição de tempo
- Como fazer?
  - Precisamos definir o que fará de uma atividade mais promissora que outra

Algoritmos Gulosos – Exemplos

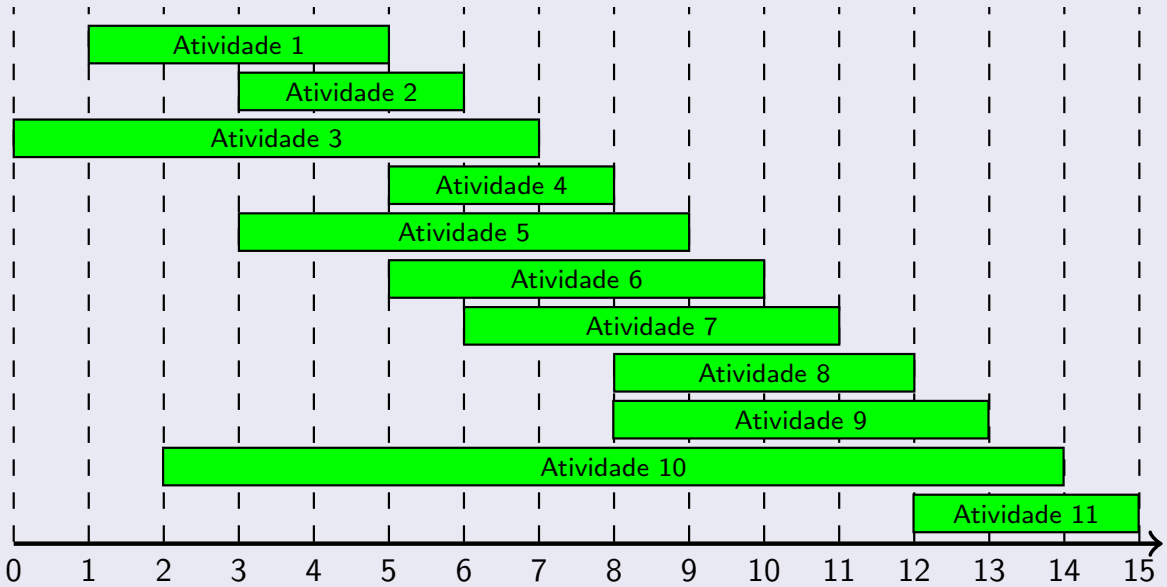
Exemplo: Seleção de Atividades

- **Tentativa 1**: Escolha primeiro as atividades que começam antes e são compatíveis

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

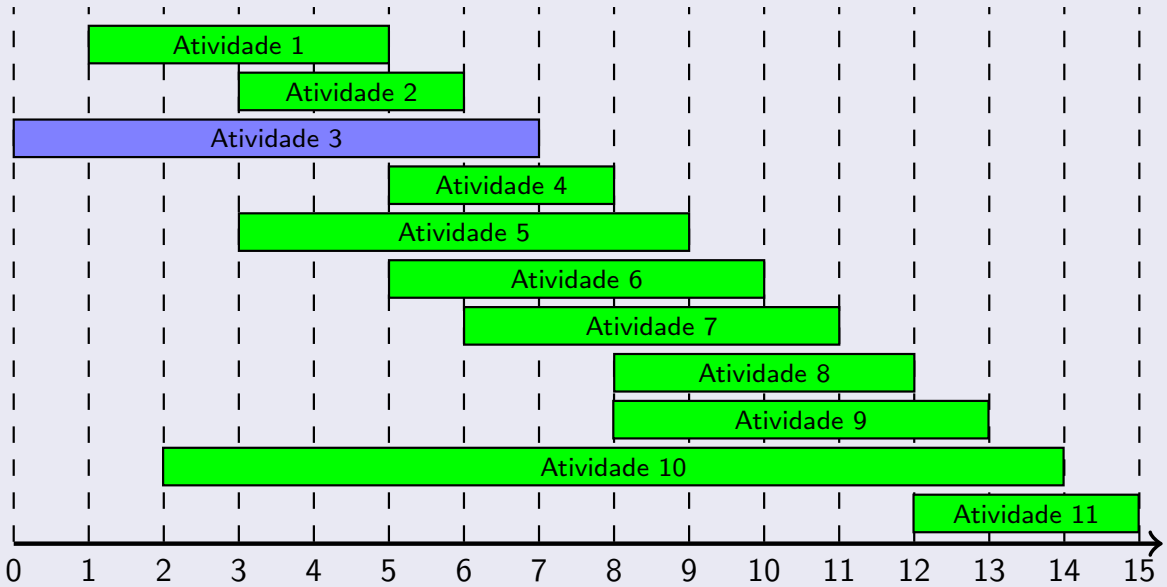
- **Tentativa 1**: Escolha primeiro as atividades que começam antes e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

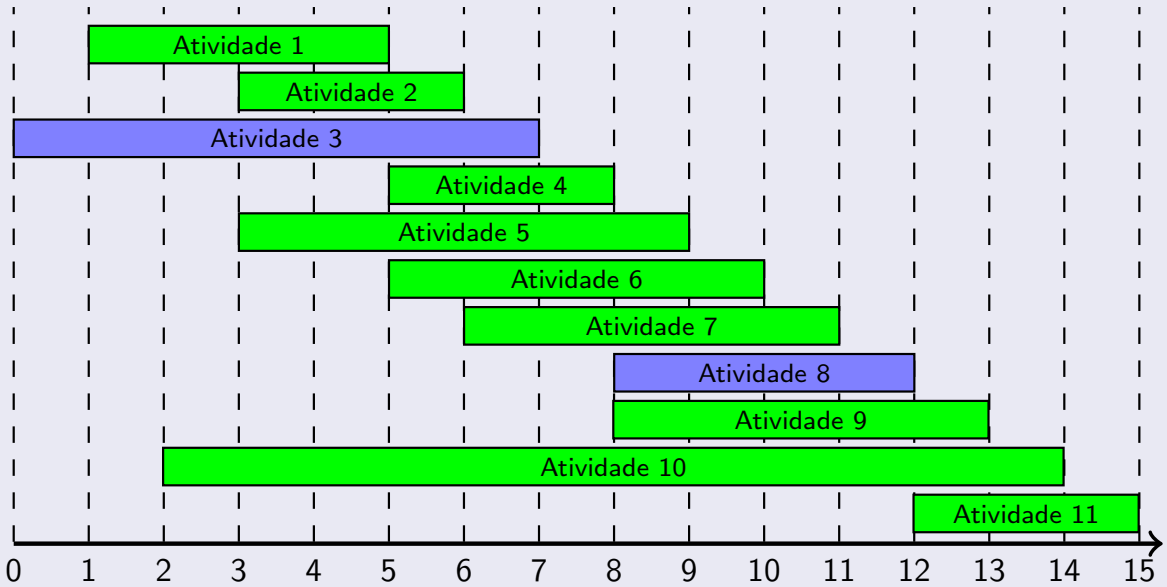
- **Tentativa 1**: Escolha primeiro as atividades que começam antes e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

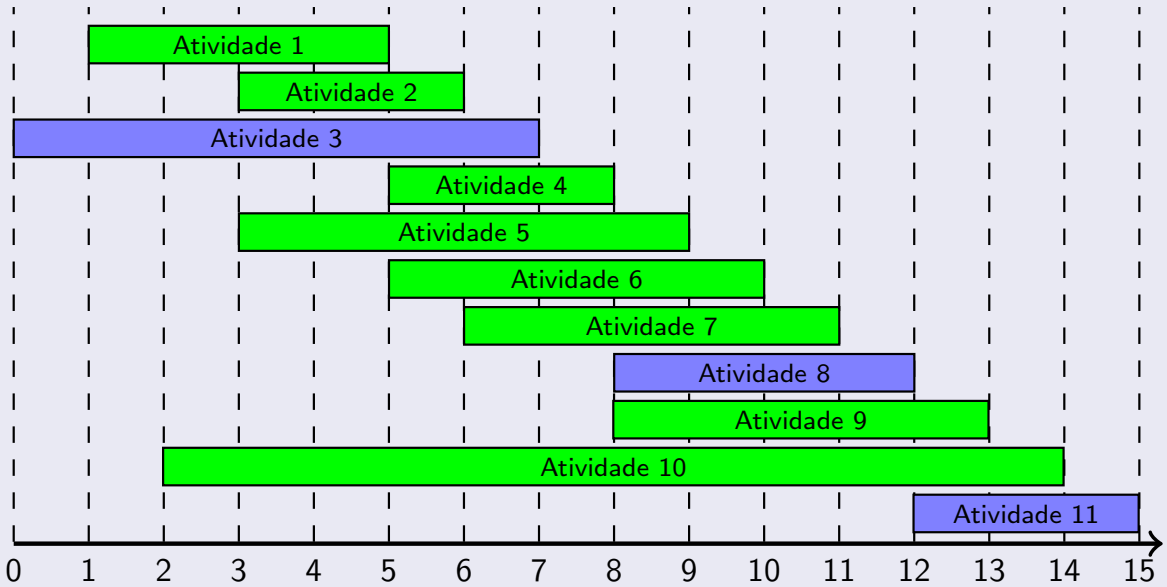
- **Tentativa 1**: Escolha primeiro as atividades que começam antes e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

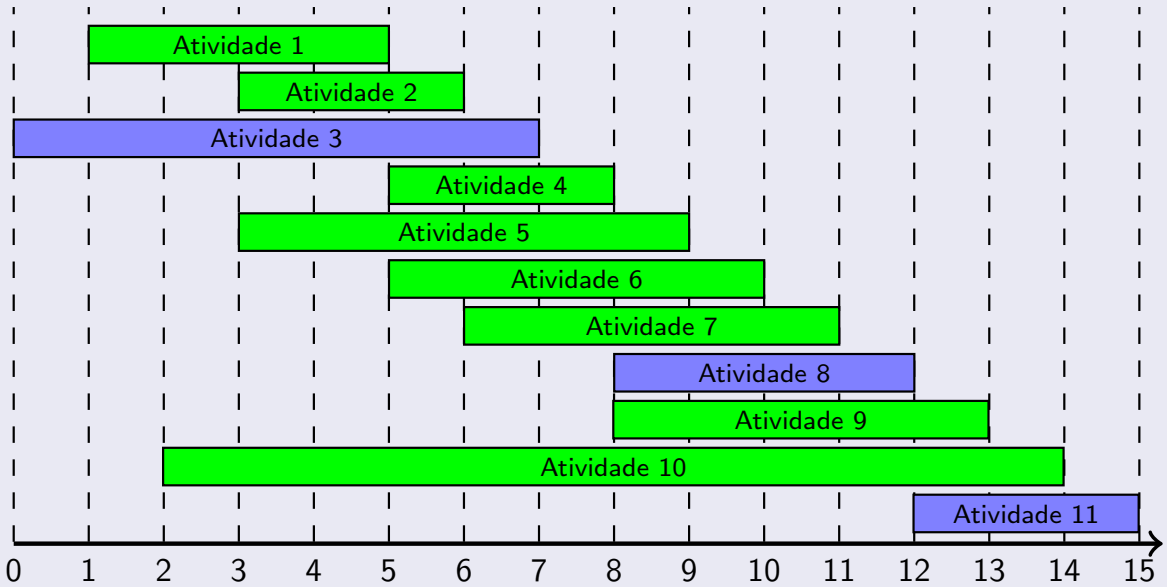
- **Tentativa 1**: Escolha primeiro as atividades que começam antes e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- **Tentativa 1**: Escolha primeiro as atividades que começam antes e são compatíveis

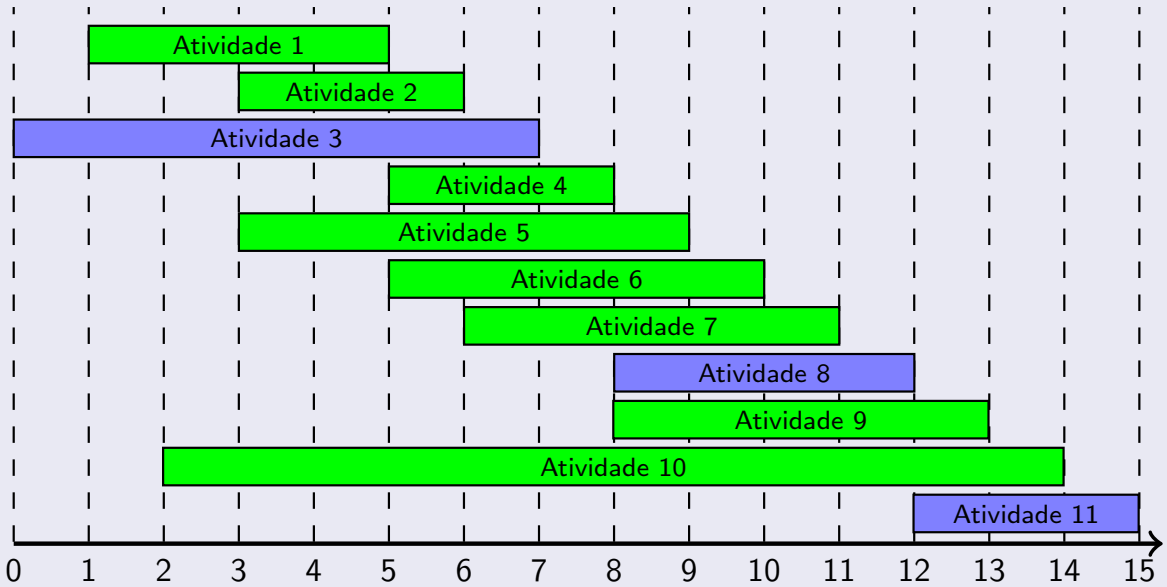


Foi boa a solução?

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- **Tentativa 1**: Escolha primeiro as atividades que começam antes e são compatíveis

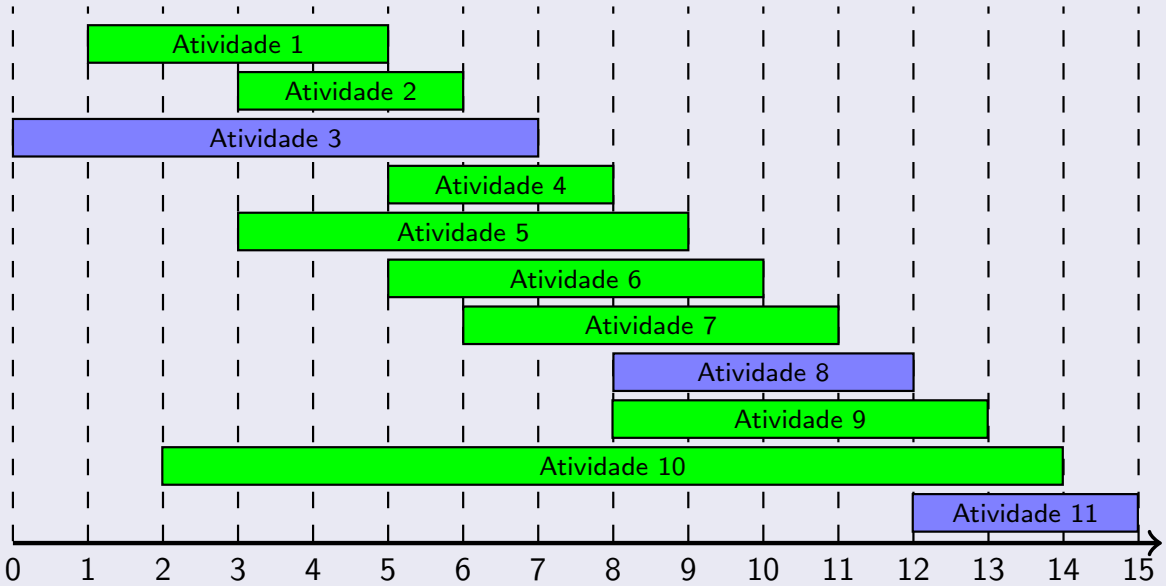


Foi boa a solução? Não realmente. Escolheu apenas as atividades 3, 8 e 11, quando poderia ter escolhido quatro (1, 4, 8 e 11)

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

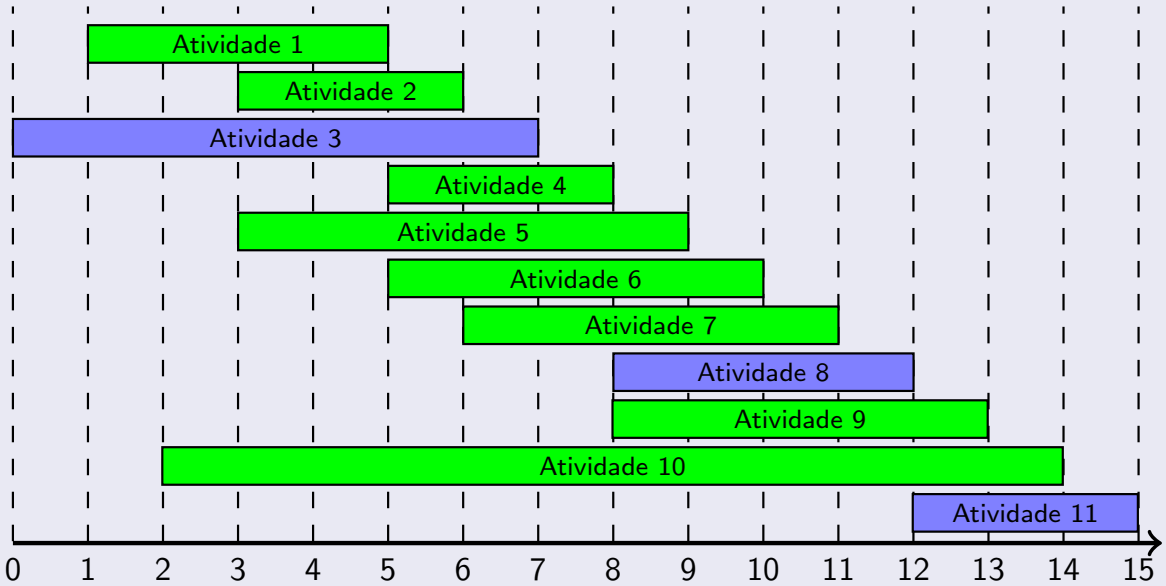
- E onde está o "guloso" nisso?



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- E onde está o "guloso" nisso?

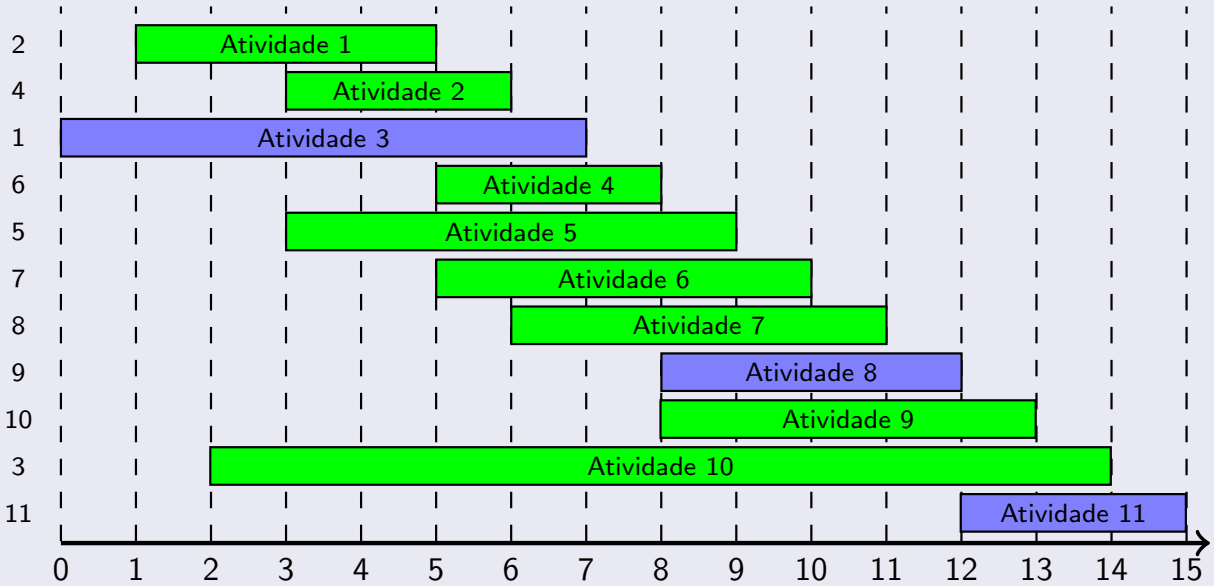


Para nossa escolha, ordenamos a entrada pelo horário de início, e pegamos as compatíveis em ordem

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- E onde está o "guloso" nisso?



Para nossa escolha, ordenamos a entrada pelo horário de início, e pegamos as compatíveis em ordem

Algoritmos Gulosos – Exemplos

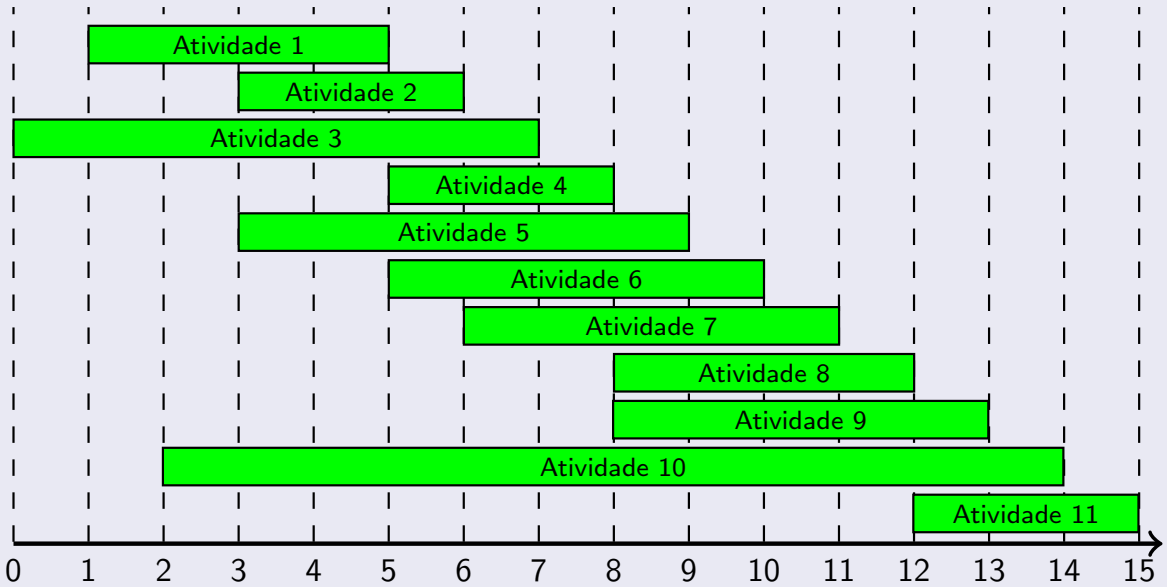
Exemplo: Seleção de Atividades

- **Tentativa 2**: Escolha primeiro as atividades que demoram menos tempo e são compatíveis

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

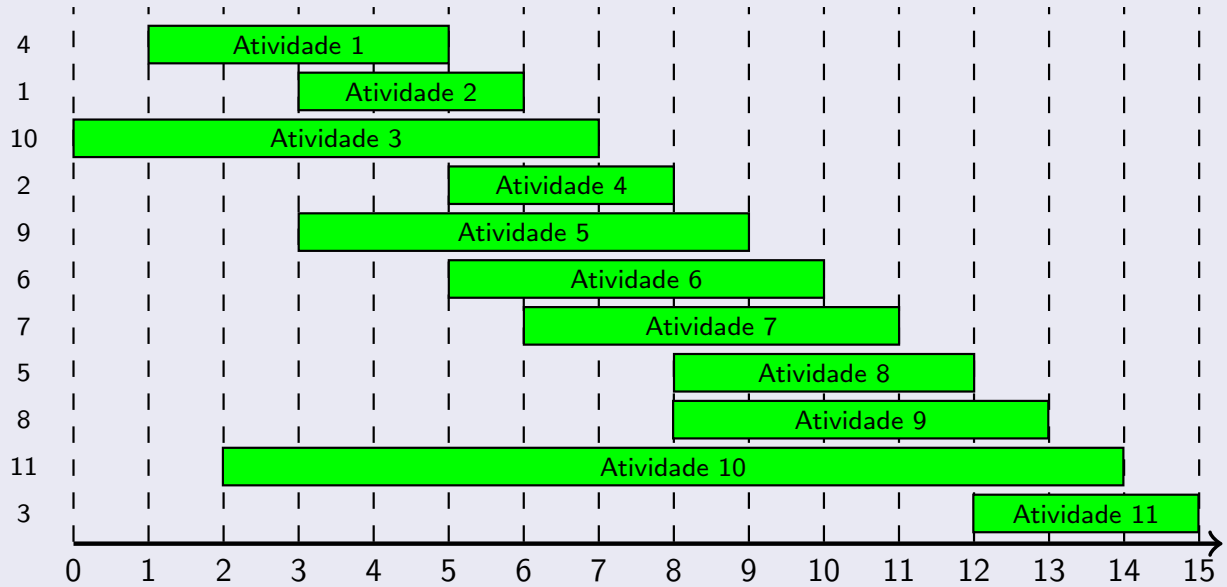
- **Tentativa 2**: Escolha primeiro as atividades que demoram menos tempo e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

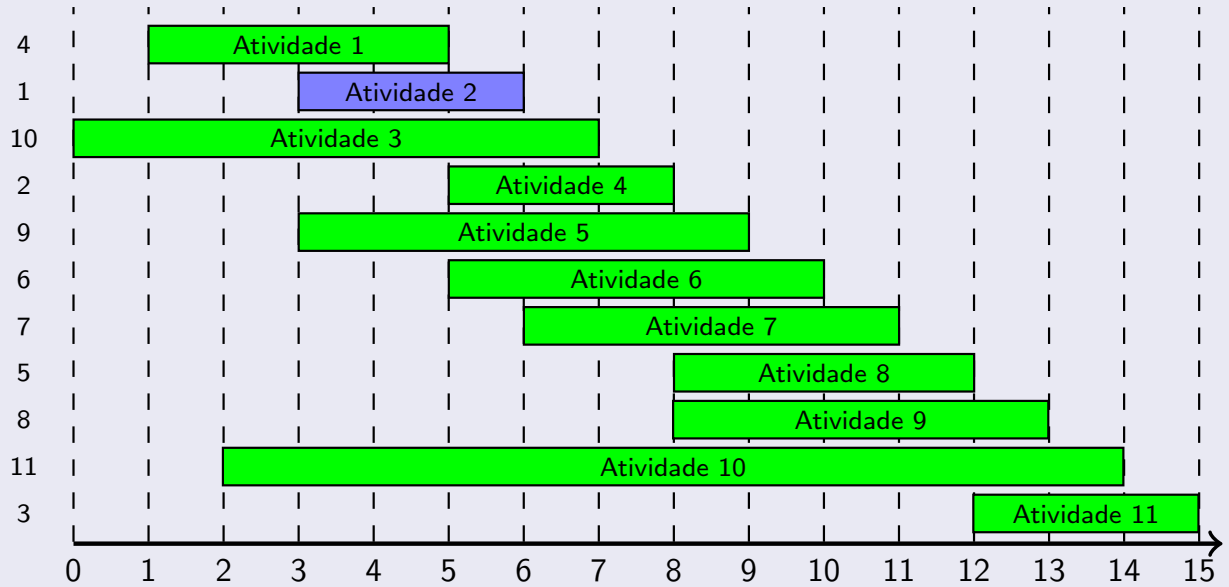
- **Tentativa 2**: Escolha primeiro as atividades que demoram menos tempo e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

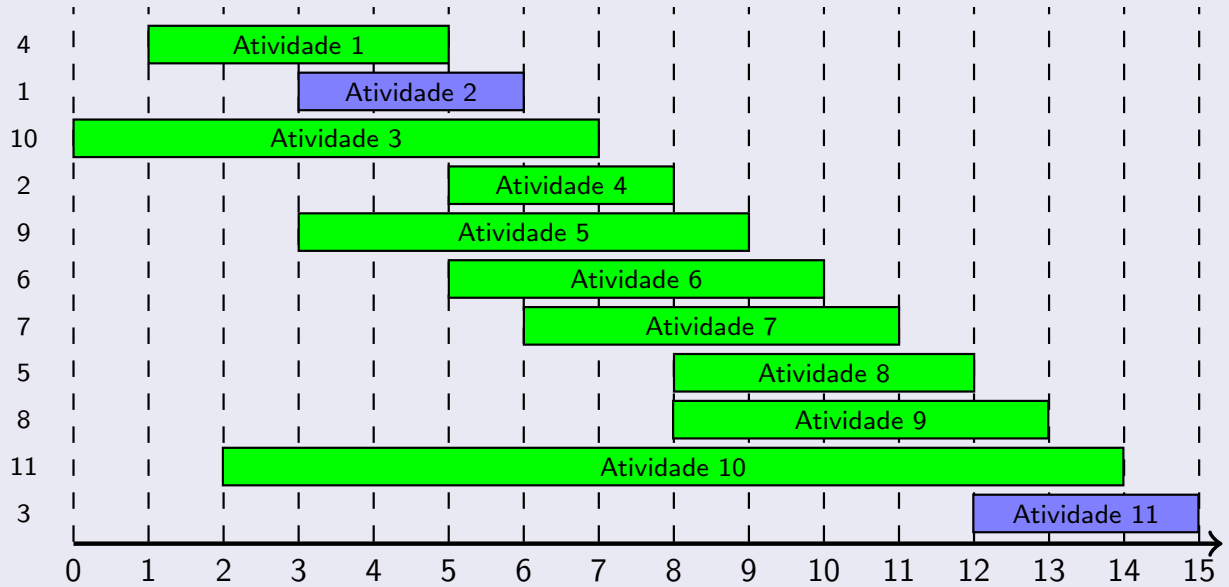
- **Tentativa 2**: Escolha primeiro as atividades que demoram menos tempo e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

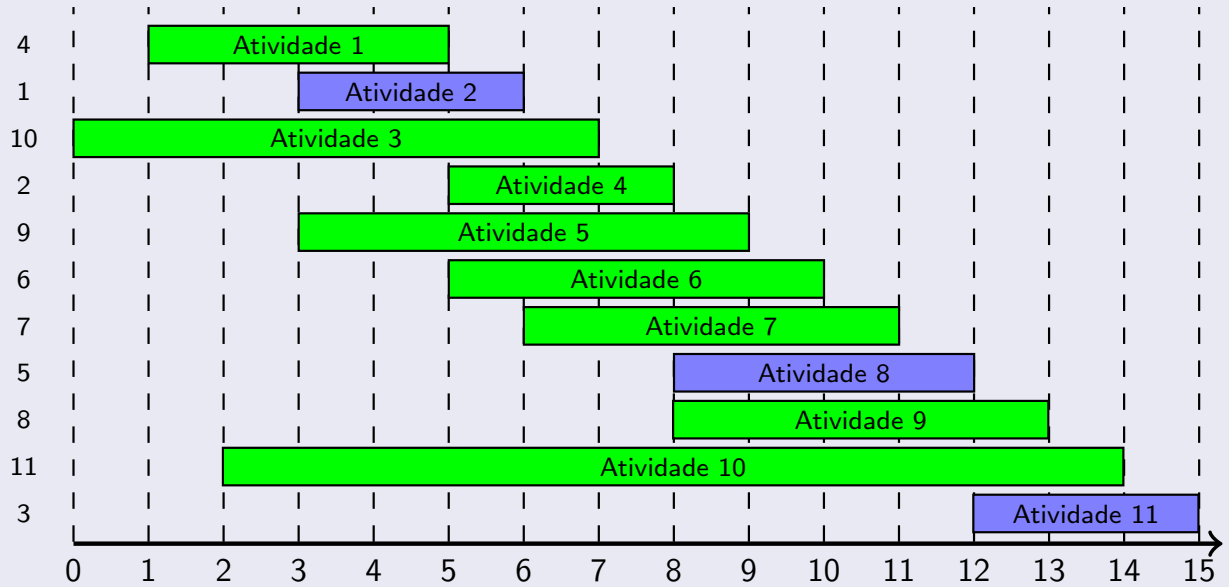
- **Tentativa 2**: Escolha primeiro as atividades que demoram menos tempo e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

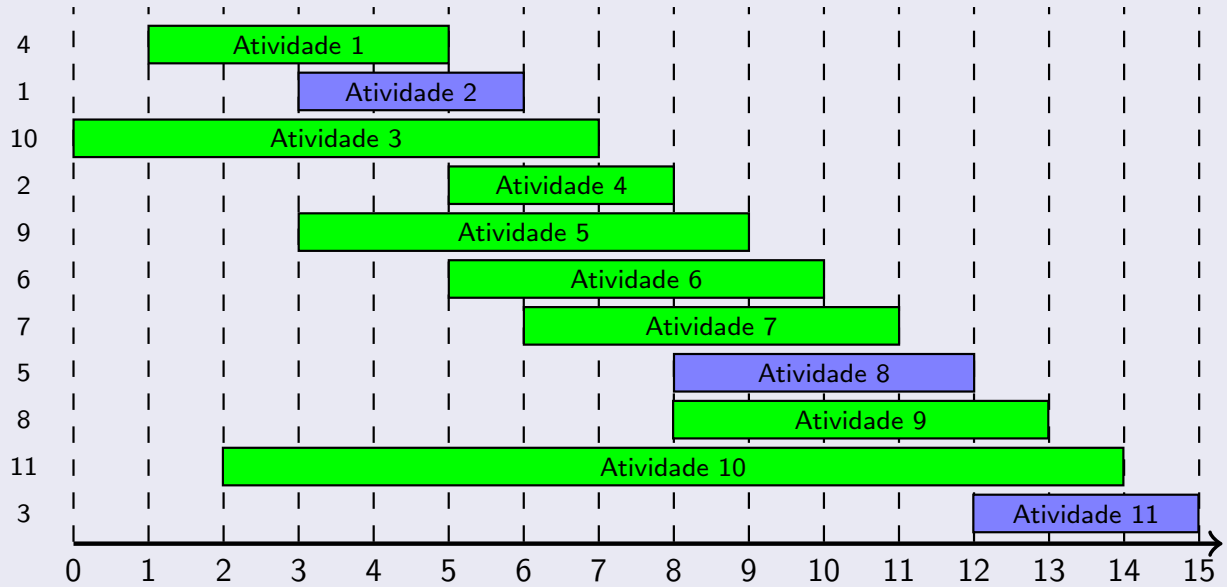
- **Tentativa 2**: Escolha primeiro as atividades que demoram menos tempo e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- **Tentativa 2**: Escolha primeiro as atividades que demoram menos tempo e são compatíveis

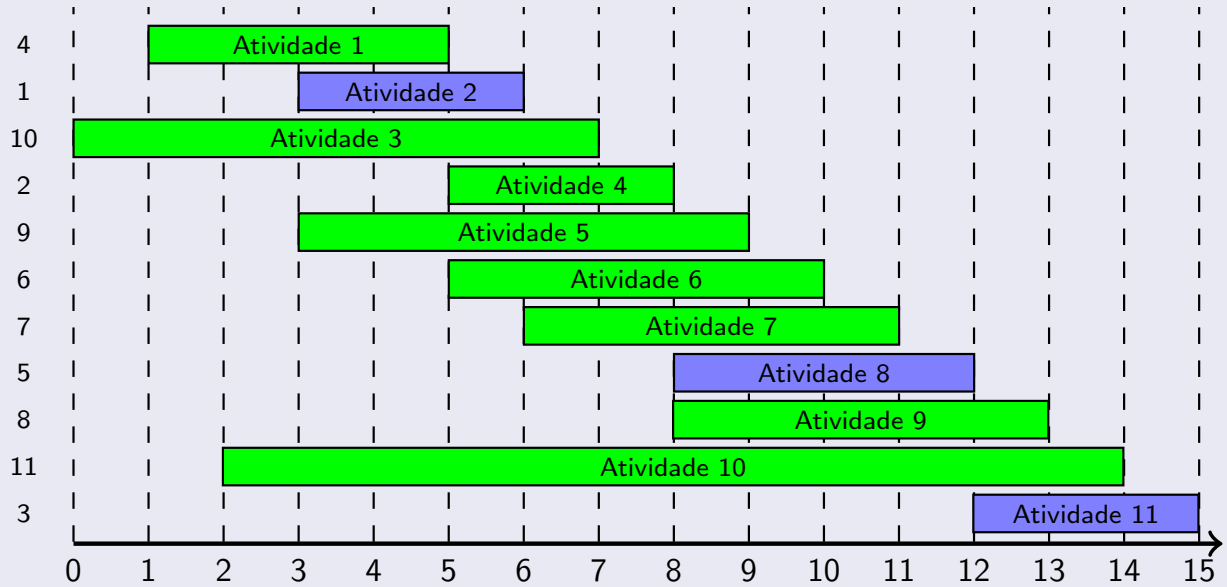


Foi boa a solução?

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- **Tentativa 2**: Escolha primeiro as atividades que demoram menos tempo e são compatíveis



Foi boa a solução? Não realmente. Escolheu apenas as atividades 2, 8 e 11, quando poderia ter escolhido quatro (1, 4, 8 e 11)

Algoritmos Gulosos – Exemplos

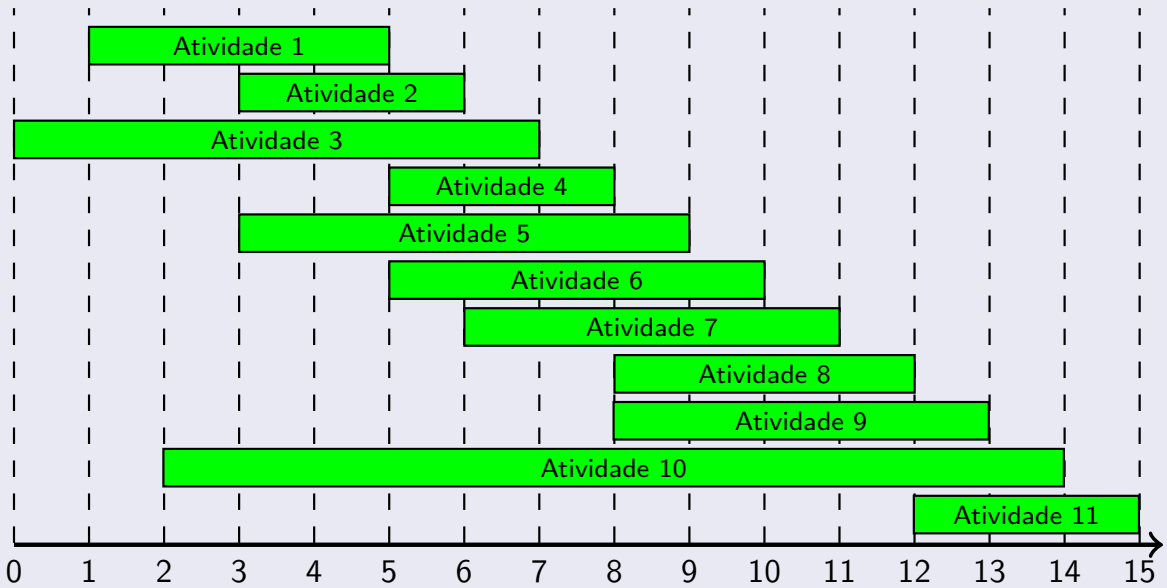
Exemplo: Seleção de Atividades

- **Tentativa 3**: Escolha primeiro as atividades que terminam antes e são compatíveis

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

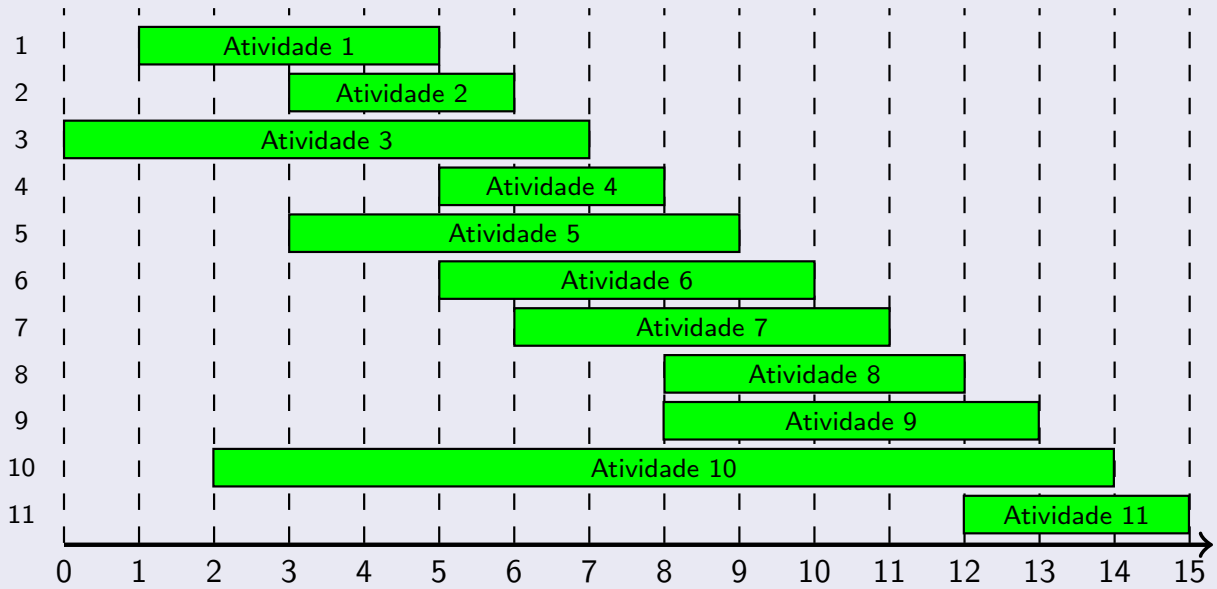
- **Tentativa 3**: Escolha primeiro as atividades que terminam antes e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

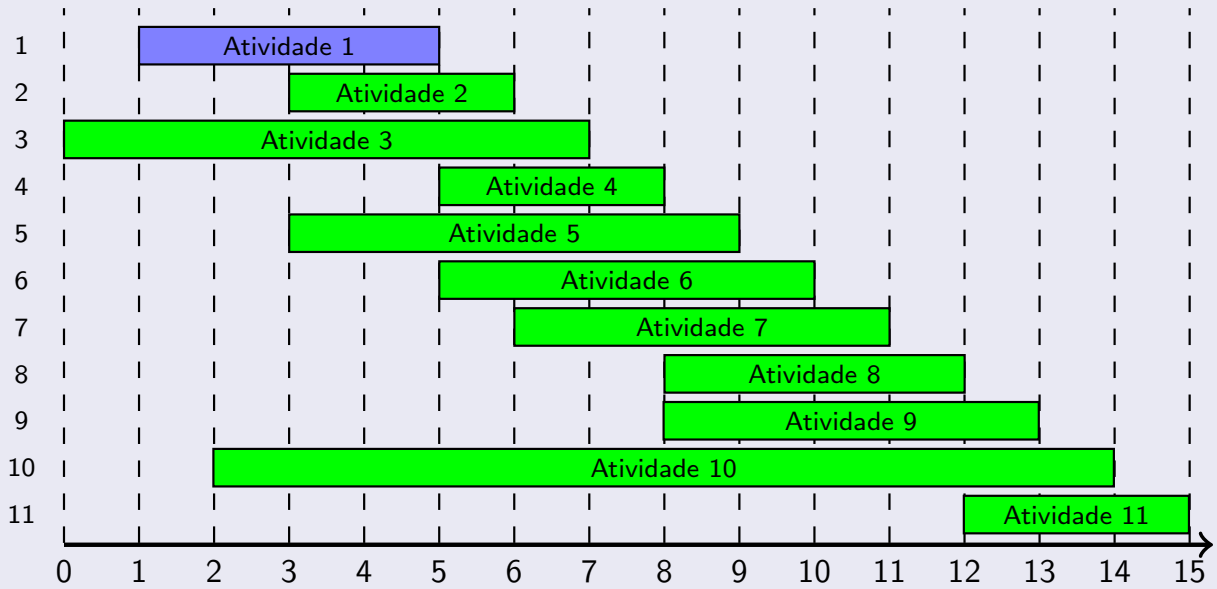
- **Tentativa 3**: Escolha primeiro as atividades que terminam antes e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

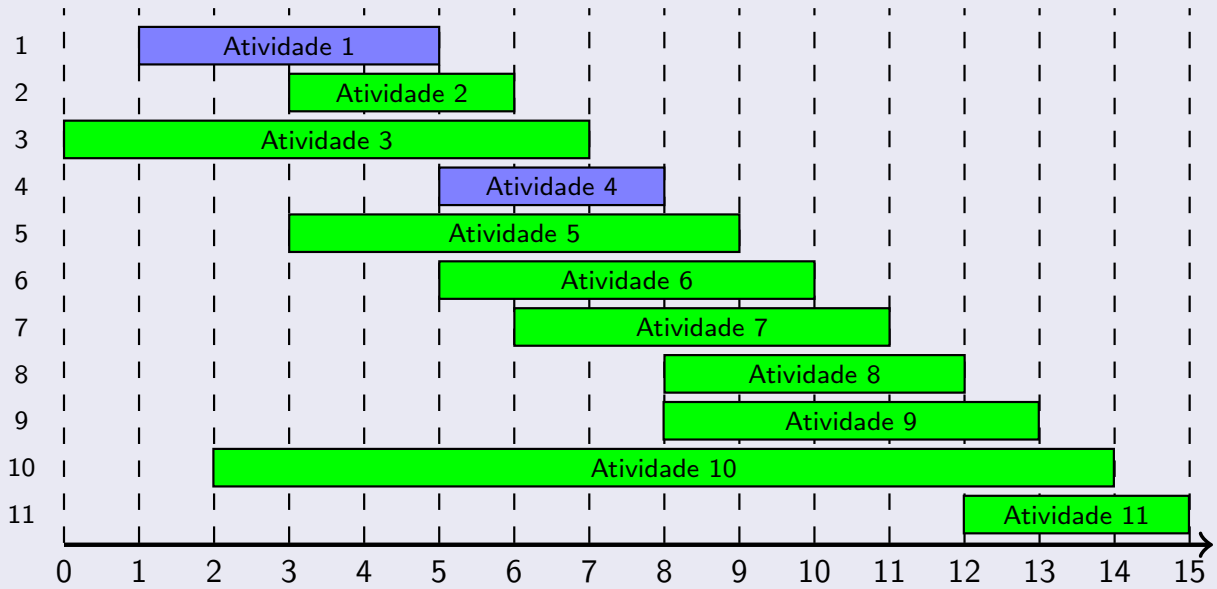
- **Tentativa 3**: Escolha primeiro as atividades que terminam antes e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

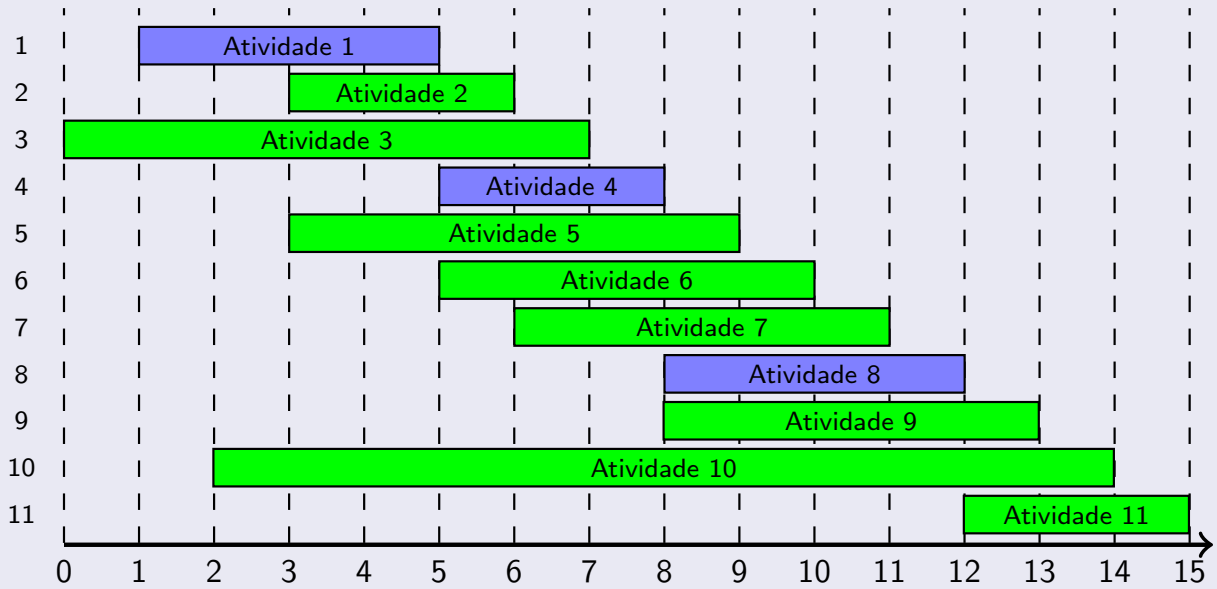
- **Tentativa 3**: Escolha primeiro as atividades que terminam antes e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

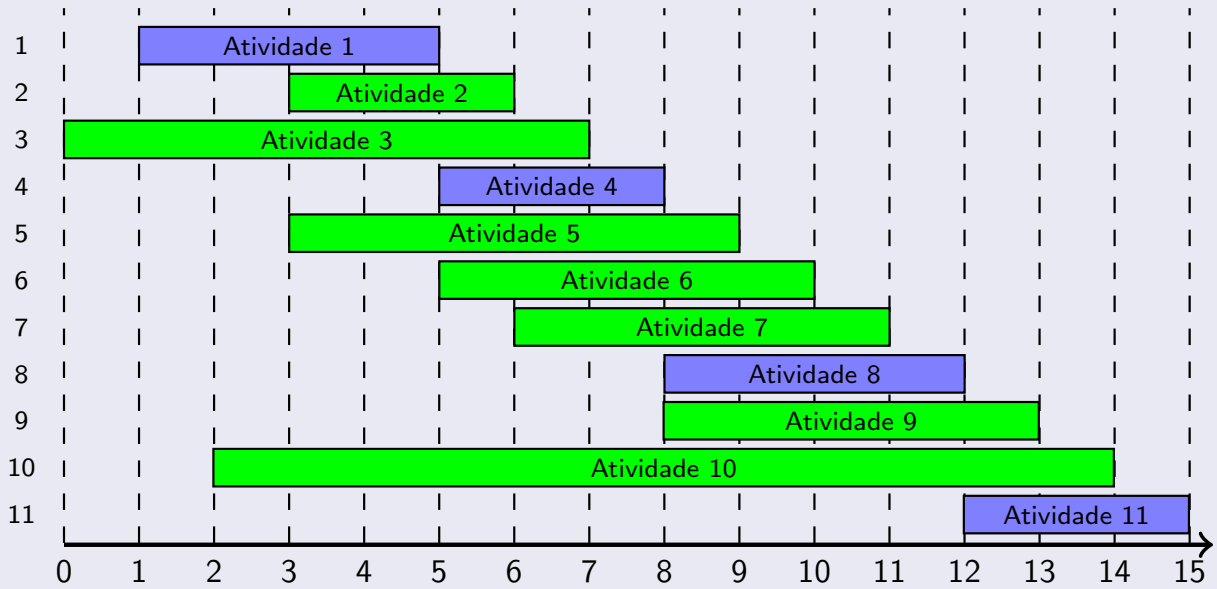
- **Tentativa 3**: Escolha primeiro as atividades que terminam antes e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

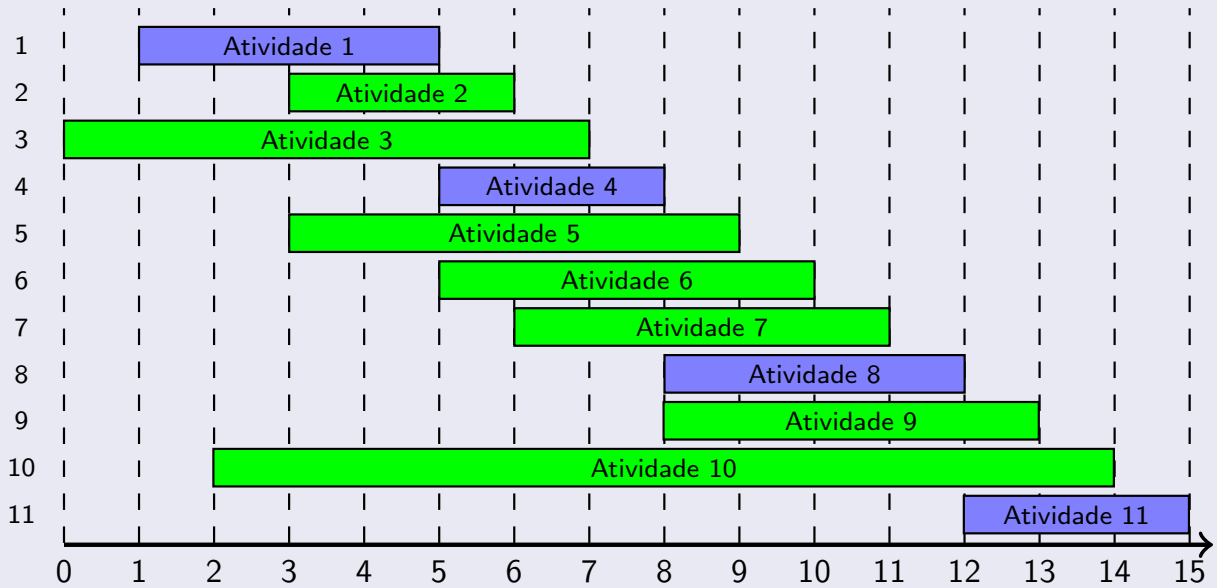
- **Tentativa 3**: Escolha primeiro as atividades que terminam antes e são compatíveis



Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- **Tentativa 3**: Escolha primeiro as atividades que terminam antes e são compatíveis

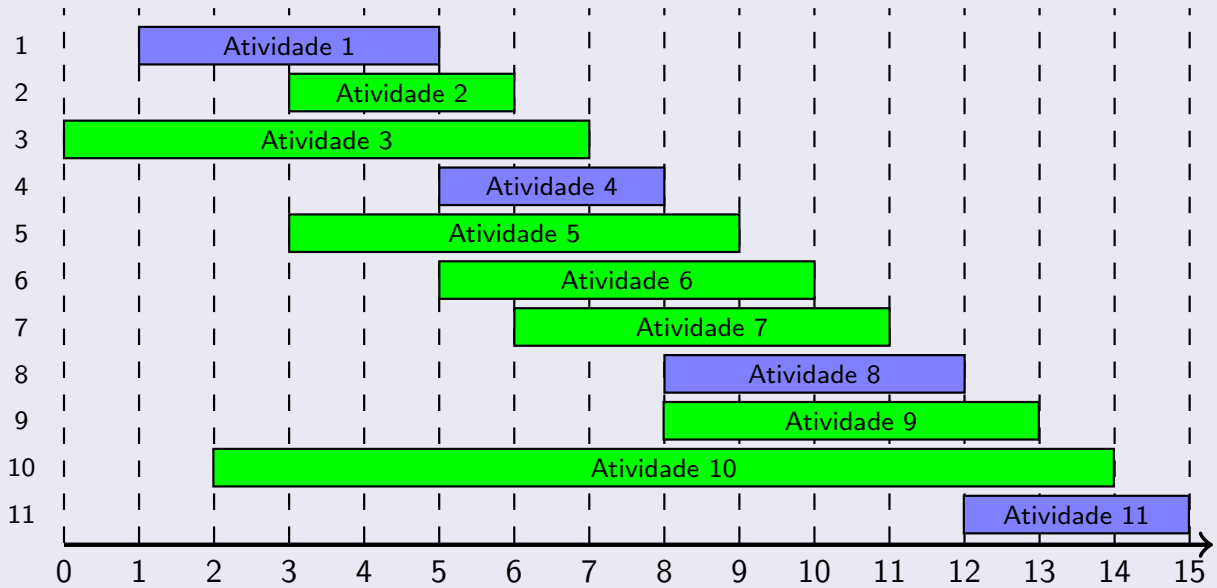


Foi boa a solução?

Algoritmos Gulosos – Exemplos

Exemplo: Seleção de Atividades

- **Tentativa 3**: Escolha primeiro as atividades que terminam antes e são compatíveis



Foi boa a solução? Agora sim. É possível demonstrar que essa solução é ótima.

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Algoritmo Guloso:

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Algoritmo Guloso:

- Entrada: lista de atividades ordenadas pelo horário de término

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Algoritmo Guloso:

- Entrada: lista de atividades ordenadas pelo horário de término
- A cada iteração:

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Algoritmo Guloso:

- Entrada: lista de atividades ordenadas pelo horário de término
- A cada iteração:
  - Verifica se a atividade atual pode ser incluída na lista de atividades

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Algoritmo Guloso:

- Entrada: lista de atividades ordenadas pelo horário de término
- A cada iteração:
  - Verifica se a atividade atual pode ser incluída na lista de atividades
- Note que a atividade atual é a que termina primeiro, dentre as atividades restantes

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Código:

```
int selecaoGulosa(int ini[], int fim[], int n) {
  int i;
  int ultimaSelecionada = 0;
  int selecionadas = 0;
  if (n==0) return 0;
  printf("a0 ");
  selecionadas++;
  for (i=1; i<n; i++)
    if (ini[i]>=fim[ultimaSelecionada]) {
      printf("a%i ", i);
      selecionadas++;
      ultimaSelecionada = i;
    }
  printf("\n");
  return selecionadas;
}
```

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Código:

```
int selecaoGulosa(int ini[], int fim[], int n) {
  int i;
  int ultimaSelecionada = 0;
  int selecionadas = 0;
  if (n==0) return 0;
  printf("a0 ");
  selecionadas++;
  for (i=1; i<n; i++)
    if (ini[i]>=fim[ultimaSelecionada]) {
      printf("a%i ", i);
      selecionadas++;
      ultimaSelecionada = i;
    }
  printf("\n");
  return selecionadas;
}
```

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Código:

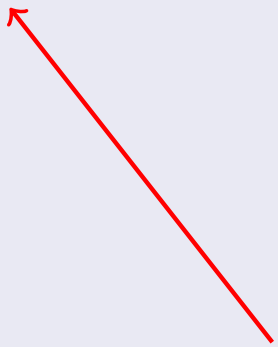
```
int selecaoGulosa(int ini[], int fim[], int n) {
  int i;
  int ultimaSelecionada = 0;
  int selecionadas = 0;
  if (n==0) return 0;
  printf("a0 ");
  selecionadas++;
  for (i=1; i<n; i++)
    if (ini[i]>=fim[ultimaSelecionada]) {
      printf("a%i ", i);
      selecionadas++;
      ultimaSelecionada = i;
    }
  printf("\n");
  return selecionadas;
}
```

Tempo de fim das atividades (em ordem crescente)

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Código:

```
int selecaoGulosa(int ini[], int fim[], int n) {
  int i;
  int ultimaSelecionada = 0;
  int selecionadas = 0;
  if (n==0) return 0;
  printf("a0 ");
  selecionadas++;
  for (i=1; i<n; i++)
    if (ini[i]>=fim[ultimaSelecionada]) {
      printf("a%i ", i);
      selecionadas++;
      ultimaSelecionada = i;
    }
  printf("\n");
  return selecionadas;
}
```



Algoritmos Gulosos – Exemplos

Seleção de Atividades – Código:

```
int selecaoGulosa(int ini[], int fim[], int n) {
  int i;
  int ultimaSelecionada = 0;
  int selecionadas = 0;
  if (n==0) return 0;
  printf("a0 ");
  selecionadas++;
  for (i=1; i<n; i++)
    if (ini[i]>=fim[ultimaSelecionada]) {
      printf("a%i ", i);
      selecionadas++;
      ultimaSelecionada = i;
    }
  printf("\n");
  return selecionadas;
}
```

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Código:

```
int selecaoGulosa(int ini[], int fim[], int n) {
  int i;
  int ultimaSelecionada = 0;
  int selecionadas = 0;
  if (n==0) return 0;
  printf("a0 ");
  selecionadas++;
  for (i=1; i<n; i++)
    if (ini[i]>=fim[ultimaSelecionada]) {
      printf("a%i ", i);
      selecionadas++;
      ultimaSelecionada = i;
    }
  printf("\n");
  return selecionadas;
}
```

Número de atividades selecionadas

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Código:

```
int selecaoGulosa(int ini[], int fim[], int n) {
  int i;
  int ultimaSelecionada = 0;
  int selecionadas = 0;
  if (n==0) return 0;
  printf("a0 ");
  selecionadas++;
  for (i=1; i<n; i++)
    if (ini[i]>=fim[ultimaSelecionada]) {
      printf("a%i ", i);
      selecionadas++;
      ultimaSelecionada = i;
    }
  printf("\n");
  return selecionadas;
}
```

A primeira atividade é sempre selecionada

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Código:

```
int selecaoGulosa(int ini[], int fim[], int n) {
  int i;
  int ultimaSelecionada = 0;
  int selecionadas = 0;
  if (n==0) return 0;
  printf("a0 ");
  selecionadas++;
  for (i=1; i<n; i++)
    if (ini[i]>=fim[ultimaSelecionada]) {
      printf("a%i ", i);
      selecionadas++;
      ultimaSelecionada = i;
    }
  printf("\n");
  return selecionadas;
}
```

Seleciona a próxima atividade compatível

Algoritmos Gulosos – Exemplos

Seleção de Atividades – Código:

```
int main() {

  int inicio[] = {1,3,0,5,3,5, 6, 8, 8, 2,12};
  int fim[] =    {4,5,6,7,8,9,10,11,12,13,14};

  int total = selecaoGulosa(inicio,fim,11);

  printf("Foram selecionadas %i atividades.\n", total);

  return 0;

}
```

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Dados:

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Dados:
  - Uma mochila que admite um certo peso

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Dados:
  - Uma mochila que admite um certo peso
  - Um conjunto de objetos, cada um com um valor e um peso

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Dados:
  - Uma mochila que admite um certo peso
  - Um conjunto de objetos, cada um com um valor e um peso
- Objetivo:

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Dados:
  - Uma mochila que admite um certo peso
  - Um conjunto de objetos, cada um com um valor e um peso
- Objetivo:
  - Selecionar o conjunto de objetos que cabem dentro da mochila de forma a maximizar o valor total dentro da mochila

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Este problema se divide em dois subproblemas distintos:

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Este problema se divide em dois subproblemas distintos:
  - Os objetos podem ser particionados (e o valor será proporcional à fração do objeto)

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Este problema se divide em dois subproblemas distintos:
  - Os objetos podem ser particionados (e o valor será proporcional à fração do objeto)
    - Ou seja, você pode colocar um pedaço do objeto dentro da mochila

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Este problema se divide em dois subproblemas distintos:
  - Os objetos podem ser particionados (e o valor será proporcional à fração do objeto)
    - Ou seja, você pode colocar um pedaço do objeto dentro da mochila
    - Ex: ouro em pó

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Este problema se divide em dois subproblemas distintos:
  - Os objetos podem ser particionados (e o valor será proporcional à fração do objeto)
    - Ou seja, você pode colocar um pedaço do objeto dentro da mochila
    - Ex: ouro em pó
    - Problema conhecido como **Problema da Mochila Fracionada**

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Este problema se divide em dois subproblemas distintos:
  - Os objetos podem ser particionados (e o valor será proporcional à fração do objeto)
    - Ou seja, você pode colocar um pedaço do objeto dentro da mochila
    - Ex: ouro em pó
    - Problema conhecido como **Problema da Mochila Fracionada**
  - Os objetos não podem ser particionados (ou estarão dentro da mochila ou fora)

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Este problema se divide em dois subproblemas distintos:
  - Os objetos podem ser particionados (e o valor será proporcional à fração do objeto)
    - Ou seja, você pode colocar um pedaço do objeto dentro da mochila
    - Ex: ouro em pó
    - Problema conhecido como **Problema da Mochila Fracionada**
  - Os objetos não podem ser particionados (ou estarão dentro da mochila ou fora)
    - Ex: ouro em barras

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila

- Este problema se divide em dois subproblemas distintos:
  - Os objetos podem ser particionados (e o valor será proporcional à fração do objeto)
    - Ou seja, você pode colocar um pedaço do objeto dentro da mochila
    - Ex: ouro em pó
    - Problema conhecido como **Problema da Mochila Fracionada**
  - Os objetos não podem ser particionados (ou estarão dentro da mochila ou fora)
    - Ex: ouro em barras
    - Problema conhecido como **Problema da Mochila Binária** ou **0-1**

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada

- Qual seria a melhor ordenação da entrada?

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso
- A solução gulosa será ótima?

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso
- A solução gulosa será ótima?
  - Sim (é demonstrável)

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso
- A solução gulosa será ótima?
  - Sim (é demonstrável)
- Algoritmo:

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso
- A solução gulosa será ótima?
  - Sim (é demonstrável)
- Algoritmo:
  1. Ordene os itens por valor/peso de modo decrescente

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso
- A solução gulosa será ótima?
  - Sim (é demonstrável)
- Algoritmo:
  1. Ordene os itens por valor/peso de modo decrescente
  2. Começando em $i = 1$, coloque na mochila o máximo do item i que puder (que existir e couber na mochila)

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso
- A solução gulosa será ótima?
  - Sim (é demonstrável)
- Algoritmo:
  1. Ordene os itens por valor/peso de modo decrescente
  2. Começando em $i = 1$, coloque na mochila o máximo do item i que puder (que existir e couber na mochila)
  3. Se ainda houver espaço na mochila, passe para o próximo item

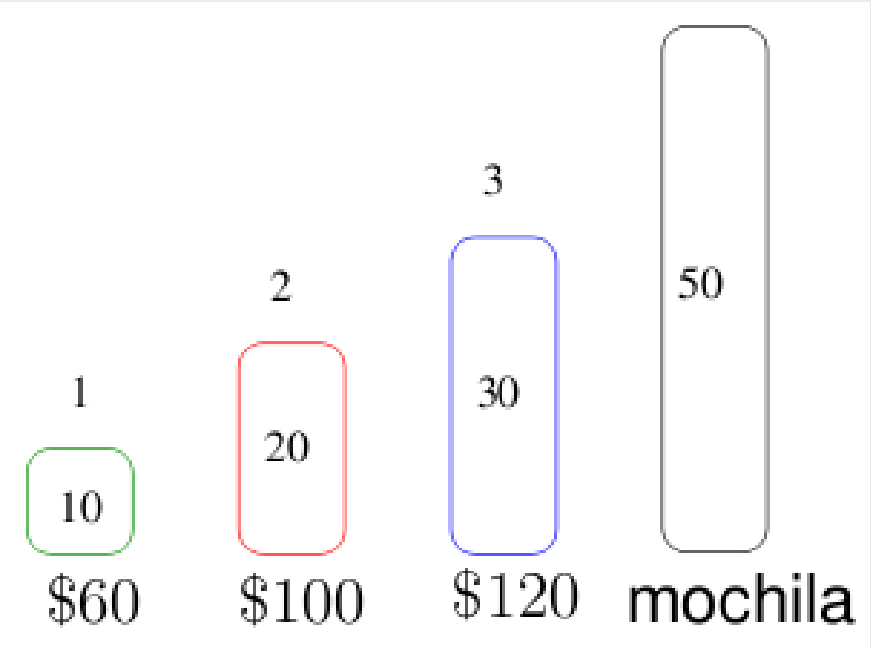
Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada

```
//P = capacidade máxima da mochila
carga = 0 //carga na mochila
i = 1
enquanto (carga < P) e (i <= n) faça {
    se (p_i <= (P - carga)
        Pegue todo o item i
    senão
        Pegue (P - carga)/p_i do item i
    Adicione a carga o peso que foi pego
    i++
}
```

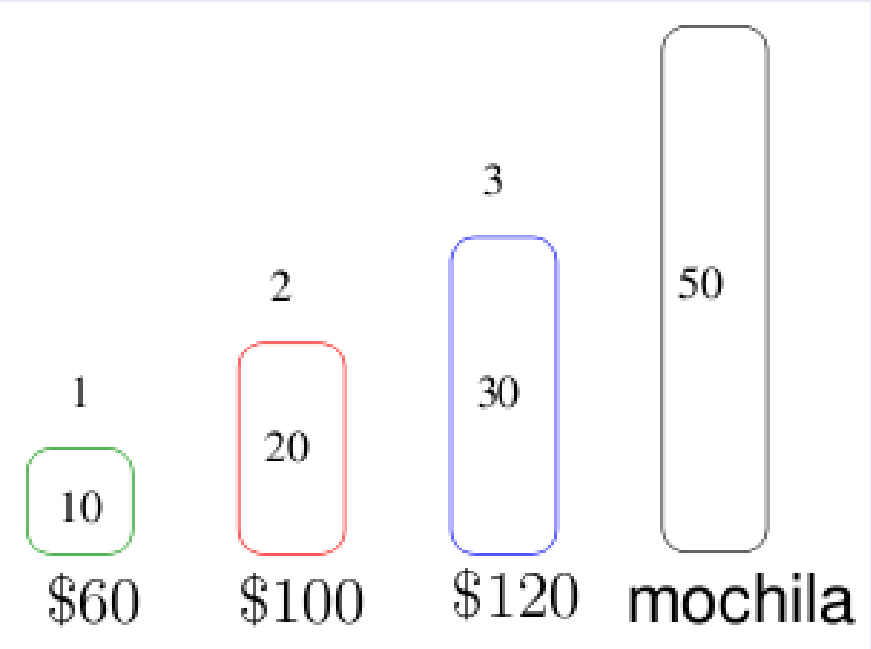
Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada



Algoritmos Gulosos – Exemplos

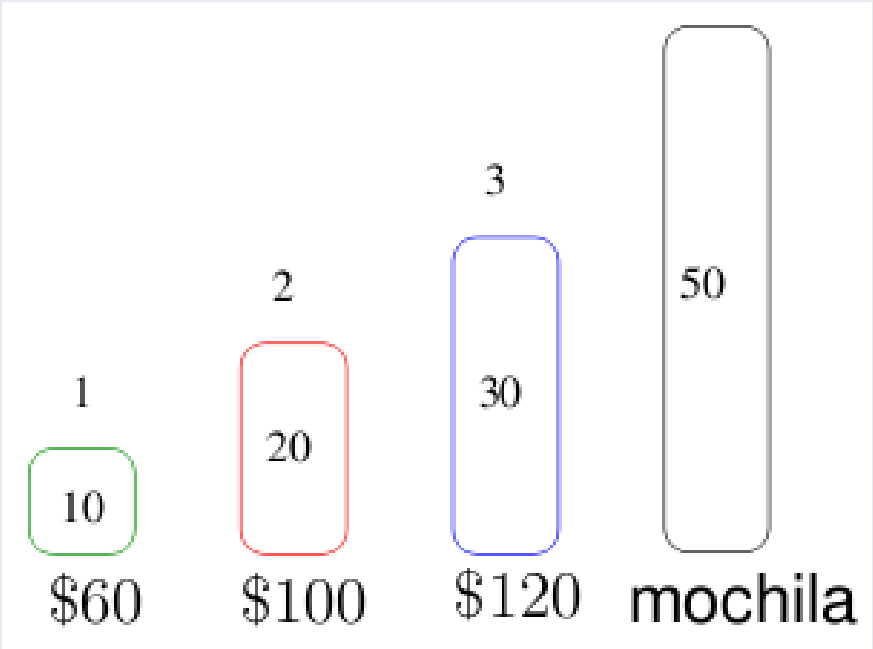
Exemplo: Problema da Mochila Fracionada



valor/peso = 6 5 4

Algoritmos Gulosos – Exemplos

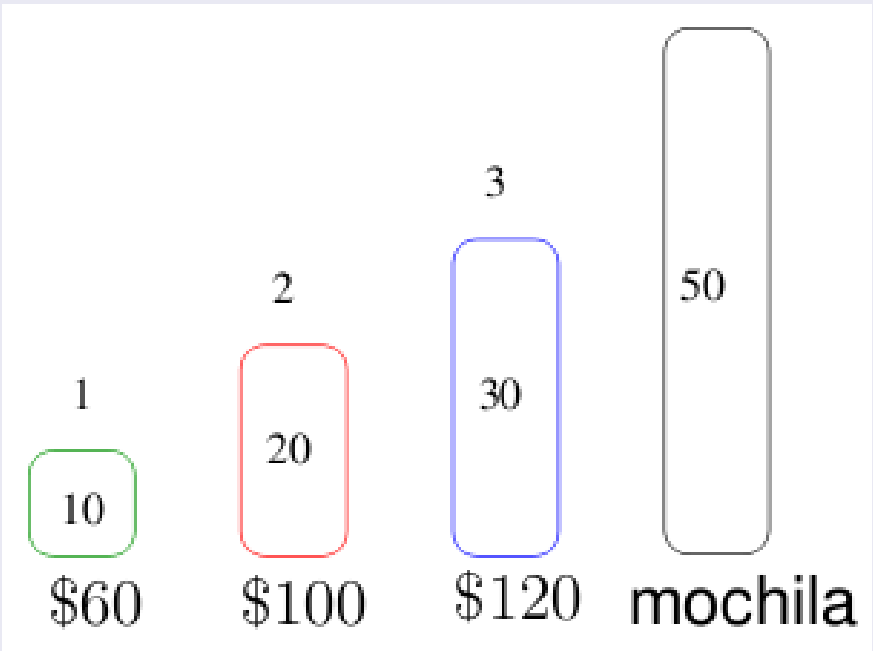
Exemplo: Problema da Mochila Fracionada

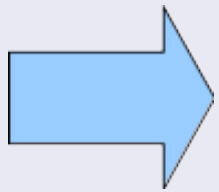


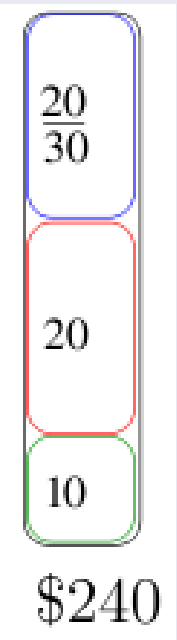
valor/peso = 6 5 4

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Fracionada







valor/peso = 6 5 4

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso
- Algoritmo:

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso
- Algoritmo:
  1. Ordene os itens por valor/peso de modo decrescente

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso
- Algoritmo:
  1. Ordene os itens por valor/peso de modo decrescente
  2. Começando em $i = 1$, coloque na mochila o máximo do item i que puder (que existir e couber na mochila)

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso
- Algoritmo:
  1. Ordene os itens por valor/peso de modo decrescente
  2. Começando em $i = 1$, coloque na mochila o máximo do item i que puder (que existir e couber na mochila)
  3. Se ainda houver espaço na mochila, passe para o próximo item

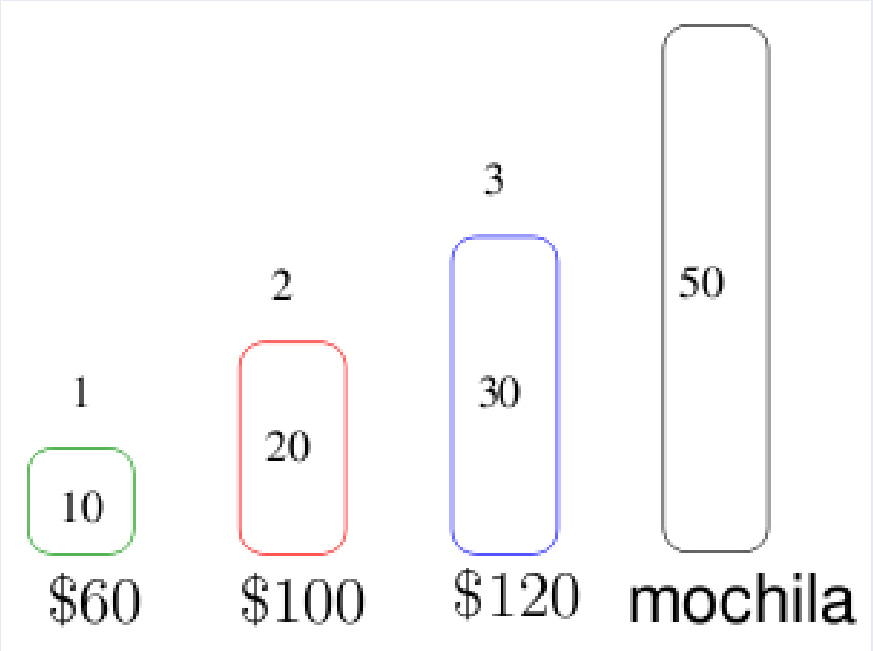
Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?
  - Ordenar por valor/peso
- Algoritmo:
  1. Ordene os itens por valor/peso de modo decrescente
  2. Começando em $i = 1$, coloque na mochila o máximo do item i que puder (que existir e couber na mochila)
  3. Se ainda houver espaço na mochila, passe para o próximo item
- A solução gulosa será ótima?

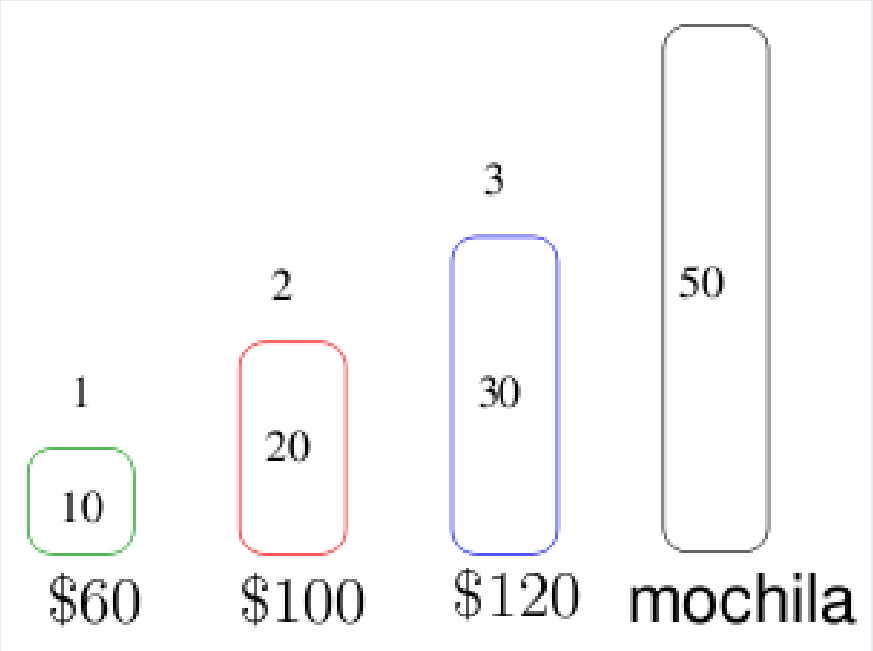
Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária



Algoritmos Gulosos – Exemplos

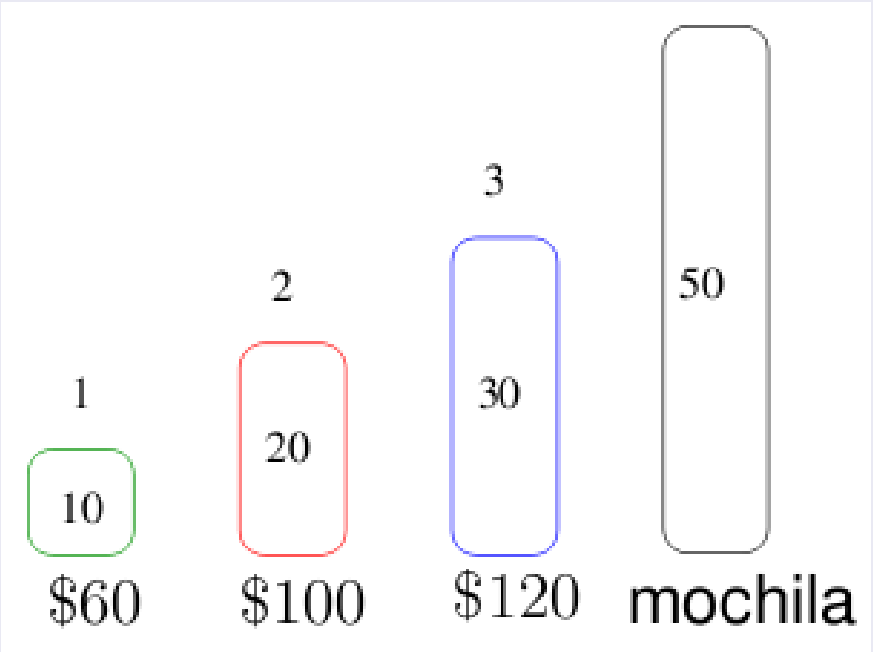
Exemplo: Problema da Mochila Binária

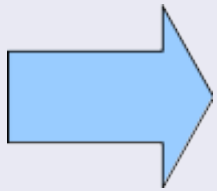


valor/peso = 6 5 4

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

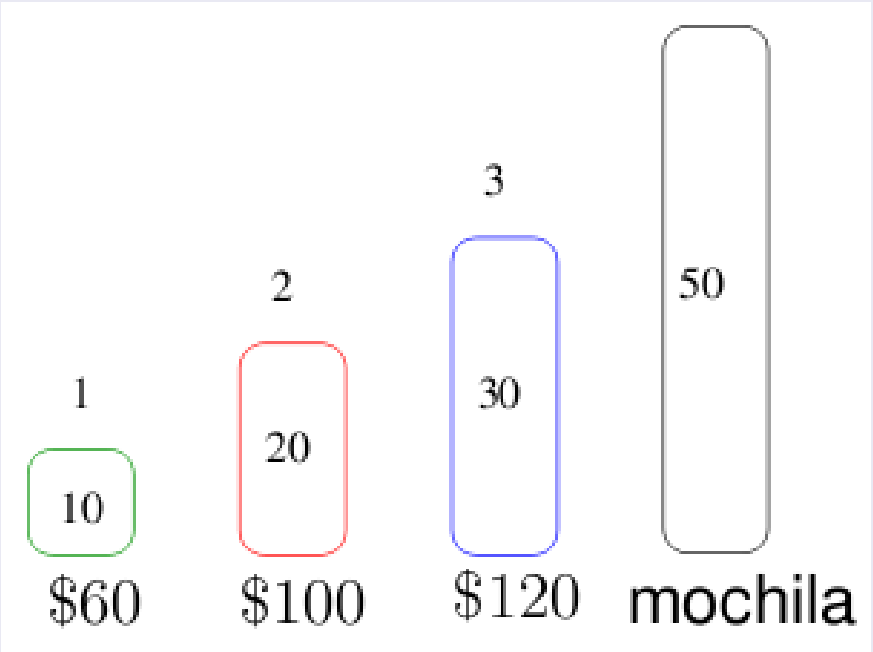


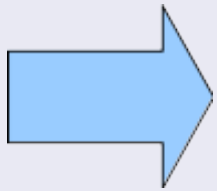


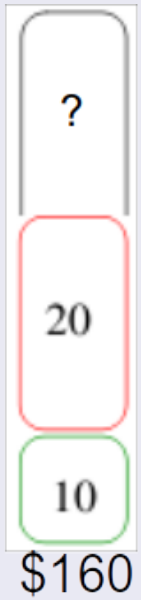
valor/peso = 6 5 4

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária



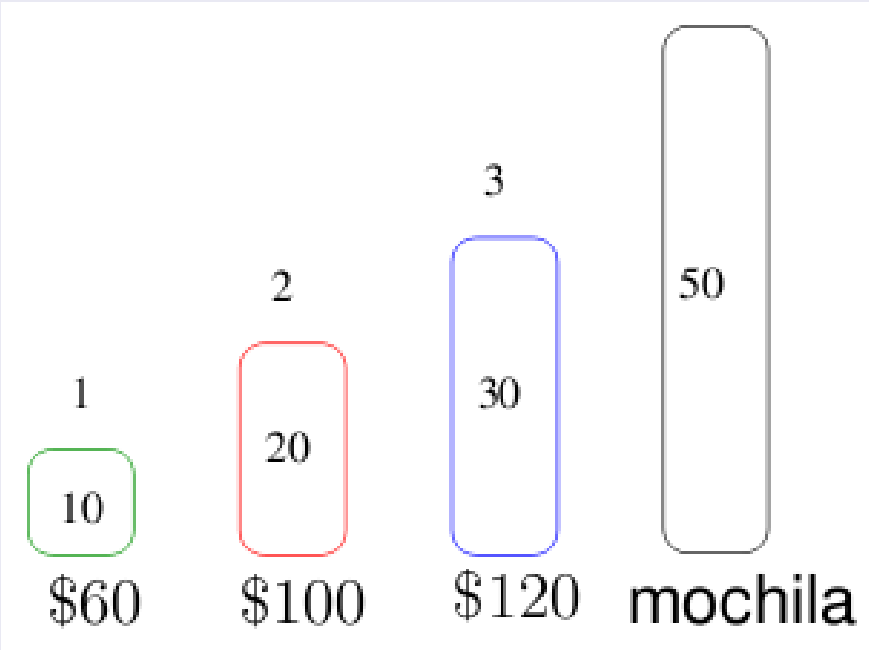


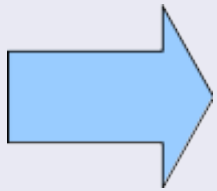


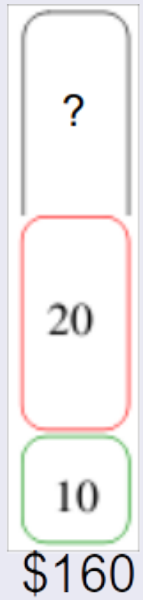
valor/peso = 6 5 4

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária







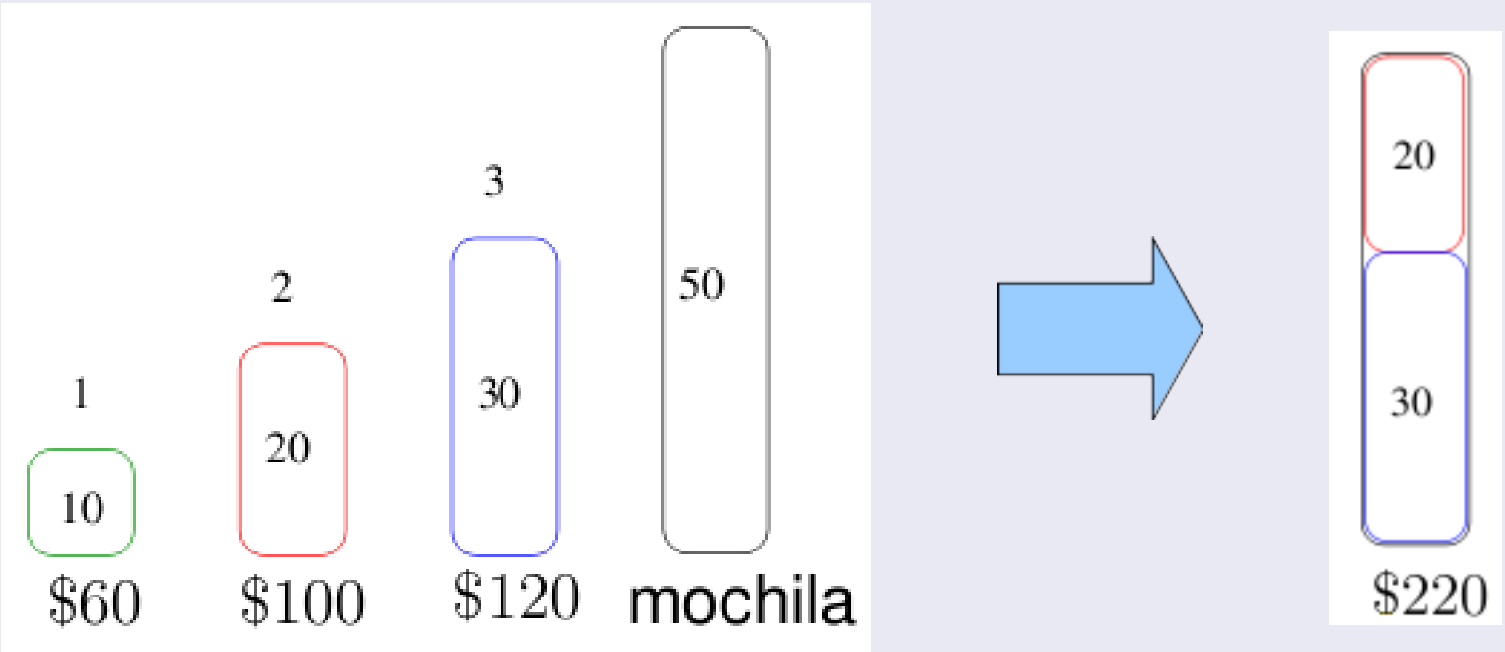
valor/peso = 6 5 4

- A solução gulosa foi ótima?

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Obviamente não. A ótima seria:



valor/peso = 6 5 4

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?
  - ~~Ordenar por valor/peso~~

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?
  - ~~Ordenar por valor/peso~~
  - Ordenar por valor de forma decrescente

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?
  - ~~Ordenar por valor/peso~~
  - ~~Ordenar pelo valor de forma decrescente~~
  - Ordenar por peso de forma crescente

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?
  - ~~Ordenar por valor/peso~~
  - ~~Ordenar pelo valor de forma decrescente~~
  - ~~Ordenar pelo peso de forma crescente~~
- Nenhuma. Não há ordenação que garanta a solução ótima com uso de um algoritmo guloso para este problema.

Algoritmos Gulosos – Exemplos

Exemplo: Problema da Mochila Binária

- Qual seria a melhor ordenação da entrada?
  - ~~Ordenar por valor/peso~~
  - ~~Ordenar pelo valor de forma decrescente~~
  - ~~Ordenar pelo peso de forma crescente~~
- Nenhuma. Não há ordenação que garanta a solução ótima com uso de um algoritmo guloso para este problema.
- Para uma solução ótima podemos usar Tentativa e Erro.

Referências

- Ziviani, Nivio. Projeto de Algoritmos: com implementações em Java e C++. Cengage. 2007.

- Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford. Introduction to Algorithms. 2a ed. MIT Press, 2001.

Aula 13 – Algoritmos: Algoritmos Gulosos

Norton T. Roman & Luciano A. Digiampietri
digiampietri@usp.br
@digiampietri

2023