



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

Elementos da estratégia gulosa

Algoritmos e Complexidade Computacional

Professor: Rennan Dantas

Universidade Federal do Ceará
Campus de Crateús

06 de janeiro de 2024

Introdução

- Um algoritmo guloso obtém uma solução ótima para um problema fazendo uma sequência de escolhas
- Para cada ponto de decisão, o algoritmo escolhe a opção que parece melhor no momento
- Essa estratégia heurística nem sempre produz uma solução ótima, mas às vezes funciona (seleção de atividades)

Introdução

- Projetamos algoritmos gulosos de acordo com a seguinte sequência de etapas:
 - 1 Expressar o problema de otimização como um problema no qual fazemos uma escolha e ficamos com um único subproblema para resolver
 - 2 Provar que sempre existe uma solução ótima para o problema original que usa a escolha gulosa, de modo que a escolha gulosa é sempre segura
 - 3 Demonstrar subestrutura ótima mostrando que, tendo feito a escolha gulosa, o que resta é um subproblema com a seguinte propriedade: se combinarmos uma solução ótima para o subproblema com a escolha gulosa que fizemos, chegamos a uma solução ótima para o problema original

Introdução

- Como saber se um algoritmo guloso resolverá determinado problema de otimização?
- Se pudermos demonstrar que o problema tem essas propriedades, estaremos no bom caminho para desenvolver um algoritmo guloso para ele

Quando usar Algoritmos Gulosos?

Quando o problema possui estrutura de subproblemas ótimos.

Quando é possível tomar decisões locais que não comprometem a solução global.

Quando o problema é de otimização e pode ser resolvido sem a necessidade de explorar todas as combinações possíveis (evitando técnicas exaustivas como força bruta).

Propriedade de escolha gulosa

- O primeiro componente fundamental é a **propriedade de escolha gulosa**: podemos montar uma solução globalmente ótima fazendo escolhas (gulosas) locais ótimas
- Em outras palavras, quando estamos considerando qual escolha fazer, escolhemos a que parecer melhor para o problema em questão, sem considerar resultados de subproblemas
- É nesse ponto que os algoritmo gulosos são diferentes da programação dinâmica
- Na programação dinâmica, fazemos uma escolha em cada etapa mas, normalmente a escolha depende das soluções dos subproblemas
- Consequentemente, em geral, resolvemos problemas de programação dinâmica de baixo para cima, passando de subproblemas menores para subproblemas maiores

Propriedade de escolha gulosa

- Em um algoritmo guloso, fazemos qualquer escolha que pareça melhor no momento e depois resolvemos o subproblema que resta
- A escolha feita por um algoritmo guloso pode depender das escolhas até o momento em questão, mas não pode depender de nenhuma escolha futura ou das soluções para subproblemas
- Assim, diferentemente da programação dinâmica, que resolve os subproblemas antes de fazer a primeira escolha, um algoritmo guloso faz sua primeira escolha antes de resolver qualquer subproblema
- Um algoritmo de programação dinâmica age de baixo para cima, ao passo que, uma estratégia gulosa em geral age de cima para baixo, fazendo uma escolha gulosa após a outra, reduzindo cada instância do problema dado a uma instância menor

Propriedade da escolha gulosa

- É claro que, temos de provar que uma escolha gulosa em cada etapa produz uma solução globalmente ótima
- Normalmente, como foi no caso do problema de seleção de atividades, a prova examina uma solução globalmente ótima para algum subproblema
- Então mostra como modificar a solução para usar a escolha gulosa no lugar de alguma outra escolha, resultando em um subproblema semelhante, porém menor

Subestrutura ótima

- Um problema exhibe **subestrutura ótima** se uma solução ótima para o problema contiver soluções ótimas para subproblemas
- Essa propriedade é um componente fundamental para avaliar a aplicabilidade da aplicação da programação dinâmica e também a de algoritmos gulosos
- Normalmente, usamos uma abordagem mais direta em relação à subestrutura ótima quando a aplicamos a algoritmos gulosos
- Podemos nos permitir o luxo de supor que chegamos a um subproblema por termos feito a escolha gulosa no problema original

Subestrutura ótima

- Na realidade, basta que demonstremos que uma solução ótima para o subproblema, combinada com a escolha gulosa já feita, produz uma solução ótima para o problema original
- Esse esquema utiliza implicitamente indução em relação aos subproblemas para provar que fazer a escolha gulosa em cada etapa produz uma solução ótima

Estratégia gulosa versus programação dinâmica

- Como as estratégias gulosas e de programação dinâmica exploram subestrutura ótima, bem que você poderia ser tentado a gerar uma solução dinâmica para um problema quando uma solução gulosa seria suficiente ou, ao contrário, achar erroneamente que uma solução gulosa funciona quando, na verdade, seria preciso uma solução de programação dinâmica
- Para ilustrar as sutilezas entre as duas técnicas, vamos investigar duas variantes de um problema clássico de otimização
- Apresentamos a seguir, o **problema da mochila 0-1**

Estratégia gulosa versus programação dinâmica

- Um ladrão que assalta uma loja encontra n itens
 - O i -ésimo item vale v_i reais e pesa w_i quilos, onde v_i e w_i são inteiros
 - O ladrão deseja pegar a carga mais valiosa possível, mas a mochila só suporta até W quilos para algum inteiro W
 - Quais itens ele deve pegar?
- Ele deve pegar os itens que lotam a mochila, mas não ultrapassam.

Critérios de um algoritmo guloso:

1° faz uma escolha rápida, sem verificar as outras opções.

2° faz uma escolha universal, para que solucione o problema de otimização o mais rápido possível.

Estratégia gulosa versus programação dinâmica

- O problema é conhecido como problema da mochila 0-1 porque, para cada item, ele deve pegá-lo ou deixá-lo para trás
- O problema da mochila fracionário tem a mesma configuração básica, mas o ladrão pode pegar frações dos itens
- Ambos os problemas exibem a propriedade da subestrutura ótima

Estratégia gulosa versus programação dinâmica

- No caso do problema 0-1, considere a carga mais valiosa, que pesa no máximo W quilos
- Se removermos o item j dessa carga, a carga restante deverá ser a carga mais valiosa que pese no máximo $W - w_j$ que o ladrão pode levar dos $n - 1$ itens originais, excluindo j
- Por comparação, no caso do problema fracionário, considere que, se removermos um peso w de um item j da carga ótima, a carga restante deverá ser a carga mais valiosa que pese no máximo $W - w$ que o ladrão pode levar dos $n-1$ itens originais mais $w_j - w$ quilos do item j

Estratégia gulosa versus programação dinâmica

- Embora os problemas sejam semelhantes, podemos resolver o problema fracionário da mochila por uma estratégia gulosa, mas não podemos usar essa mesma estratégia para resolver o problema 0-1
- Para resolver o problema fracionário, primeiro calculamos o valor por quilo v_i/w_i para cada item
- Obedecendo a uma estratégia gulosa, o ladrão começa levando o máximo possível do item que tenha o maior valor por quilo
- Se o suprimento desse item se esgotar e o ladrão ainda puder carregar mais, levará o máximo possível do próximo item que tenha o maior valor por quilo e assim por diante até alcançar a capacidade W
- Portanto, ordenando os itens por valor por quilo, o algoritmo guloso é executado no tempo $O(n \lg n)$

Estratégia gulosa versus programação dinâmica

- Para ver que essa estratégia gulosa não funciona para o problema da mochila 0-1, considere a instância ilustrada na Figura (a)
- Nesse exemplo, a estratégia levaria o item 1 primeiro, pois é o que possui o maior valor por quilo
- Porém, a solução ótima deveria levar os itens 2 e 3 e deixar o item 1 para trás
- O item 1 não pertence a nenhuma solução ótima

Estratégia gulosa versus programação dinâmica

- No problema da mochila 0-1, quando consideramos incluir um item na mochila, temos de comparar a solução para o subproblema que inclua tal item com a solução para o subproblema que exclua esse mesmo item, antes de fazermos a escolha
- O problema formulado dessa forma dá origem a muitos subproblemas sobrepostos - uma marca registrada da programação dinâmica
- Realmente, podemos usar programação dinâmica para resolver o problema da mochila 0-1

Elementos da estratégia gulosa

Estratégia gulosa versus programação dinâmica

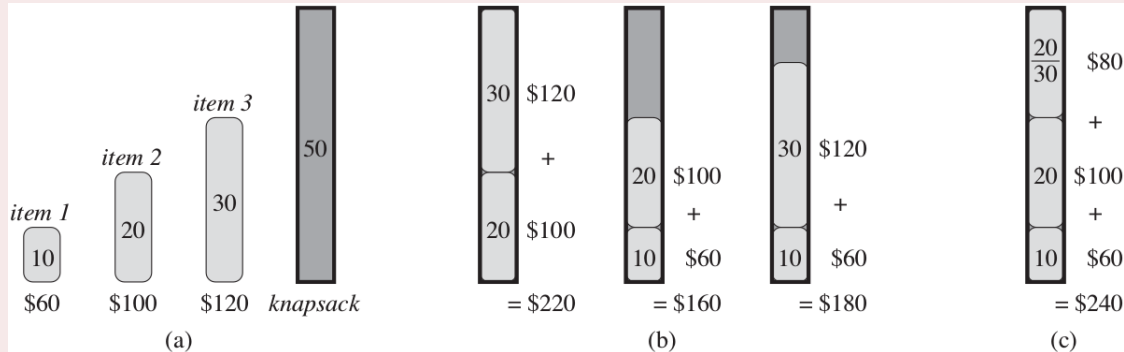


Figura: Fonte: Livro Algoritmos - Cormen

O que vem por aí?

- Códigos de Huffman



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

Elementos da estratégia gulosa

Algoritmos e Complexidade Computacional

Professor: Rennan Dantas

Universidade Federal do Ceará
Campus de Crateús

06 de janeiro de 2024