

Paradigmas de Resolução de Problemas

Overview

Prof. Edson Alves – UnB/FGA

Paradigmas de Solução de Problemas

- Os quatro principais paradigmas para solução de problemas são:
 1. Busca completa
 2. Divisão e conquista
 3. Guloso
 4. Programação Dinâmica
- Os paradigmas diferem entre si nas formas em que avaliam o espaço de soluções e como procedem quando estão defronte a mais de uma possibilidade
- A escolha do paradigma correto para a solução de um problema impactará na dificuldade da escrita do código e na complexidade assintótica do algoritmo

Busca Completa

- A busca completa avalia, um a um, todos os candidatos à solução do problema
- Quando um algoritmo de busca completa se encontra em uma situação com duas ou mais alternativas, ele avaliará todas elas
- Por conta destas características, estes algoritmos tendem a ter as maiores complexidades, quando comparados com algoritmos baseados nos outros paradigmas
- Algoritmos de busca completa tendem a ter implementações simples e que produzem a saída correta para todas as entradas possíveis
- Contudo podem ser viáveis apenas para entradas relativamente pequenas

Divisão e Conquista

- No paradigma da divisão e conquista um problema é subdividido em problemas menores
- Se estes problemas menores ainda não tem solução óbvia, eles são novamente subdivididos, até que possam ser resolvidos trivialmente
- Em seguida, as soluções dos subproblemas devem ser combinadas para formar a solução do problema original
- A natureza do paradigma leva a implementações, em geral, recursivas
- A complexidade dos algoritmos baseadas em divisão e conquista não são óbvias e dependem da relação de recorrência entre o número de passos necessários para a solução do problema original e dos subproblemas, para a divisão e para a combinação das soluções

- Um algoritmo guloso, quando confrontado com múltiplas alternativas, escolhe aquela que é localmente ótima, com a expectativa que estas escolhas levem à solução ótima global
- Em geral, algoritmos gulosos são os que tem menor complexidade assintótica, quando comparados com algoritmos baseados nos outros paradigmas
- Contudo, algoritmos gulosos tendem a não produzirem a saída correta para todas as entradas, se não for tomado o devido cuidado
- Para que sejam corretos é preciso provar que a estratégia de escolha efetivamente produz a solução ótima

Programação Dinâmica

- Assim como na busca completa, a programação dinâmica avalia todos os candidatos à solução e todas as alternativas possíveis
- A diferença é que a programação dinâmica evita que o algoritmo compute um mesmo problema múltiplas vezes, por meio de uma técnica denominada memorização
- Se o problema tem subestrutura ótima (isto é, as soluções ótimas dos subproblemas podem ser combinadas para a solução ótima do problema) e os subproblemas se repetem, este problema é candidato a ter uma solução que utiliza programação dinâmica
- As implementações são praticamente idêntica às baseadas em busca completa, exceto pela memorização

1. **CORMEN**, Thomas H.; **LEISERSON**, Charles E.; **RIVEST**, Ronald; **STEIN**, Clifford. *Introduction to Algorithms*, 3rd Edition, MIT Press, 2009.
2. **LAARKSONEN**, Antti. *Competitive Programmer's Handbook*, 2017.
3. **HALIM**, Steve; **HALIM**, Felix. *Competitive Programming 3*, Lulu, 2013.