UNIVERSIDADE FEDERAL DE OURO PRETO
Rafael Francisco de Oliveira - 2021.10171
PCC104 - Projeto e Análise de Algoritmos
Github: raffoliveira
Lista 6

---

1. Programação Dinâmica

   (a) Para que tipo de problema serve a técnica de Programação Dinâmica?

   A programação dinâmica é uma técnica para resolver problemas com subproblemas sobrepostos. Normalmente, esses subproblemas surgem de uma recorrência relacionando a solução de um determinado problema às soluções de seus subproblemas menores. Em vez de resolver subproblemas sobrepostos, repetidamente, a programação dinâmica sugere resolver cada um dos subproblemas menores apenas uma vez e registrar os resultados em uma tabela a partir da qual uma solução para o problema original pode então ser obtida. Logo, se o problema principal puder ser dividido em subproblemas menores e esses subproblemas puderem ser combinados para chegar na solução desejada, esse tipo de técnica pode ser usada.

2. Backtracking

   (a) Descreva a técnica de backtracking.

   Backtracking é uma técnica de desenvolvimento de algoritmo inteligente da técnica busca exaustiva. O princípio é construir as soluções em forma de árvore (desenvolvida em formato de *deep-first*), uma componente de cada vez e avaliando parcialmente cada possível candidato a solução do problema ao mesmo tempo. Se a solução parcial construída puder continuar a ser construída sem violar nenhuma restrições do problema, então isto é feito pegando o primeira opção dos componentes da solução. Se o próximo elemento não formar uma solução parcial completa, um próximo elemento é escolhido até que as opções acabem. Quando não houver mais opção, e nenhuma possível solução for encontrada, a técnica volta um nível na construção da solução possível e explora as opções disponíveis. O algoritmo para quando encontra uma solução completa ou as soluções completas possíveis (depende do tipo de problema).

3. Branch and Bound

(a) Descreva a técnica de branch and bound. Faça um comparação com a técnica de backtracking.

Branch and bound é uma técnica de desenvolvimento de algoritmo inteligente da técnica busca exaustiva focada em problemas de otimização. O princípio é construir as soluções em forma de árvore (desenvolvida em formato de *deep-first*), uma componente de cada vez e avaliando parcialmente cada possível candidato a solução do problema ao mesmo tempo (chamada de **feasible solution**).

A cada iteração, a **feasible solution** e solução ótima (chamada de **optimial solution** - uma feasible solution com o melhor valor objetivo do problema - *bound*) são avaliadas. A primeira se satisfaz as restrições do problema e a segunda se é uma solução ótima baseada no limite (**bound**) do problema.

Se a solução parcial construída puder continuar a ser construída sem violar nenhuma restrições do problema, então isto é feito pegando o primeira opção dos componentes da solução. Se o próximo elemento não formar uma solução parcial completa, um próximo elemento é escolhido até que as opções acabem. Quando não houver mais opção, e nenhuma possível solução for encontrada, a técnica volta um nível na construção da solução possível e explora as opções disponíveis. O algoritmo para quando encontra uma solução completa ou as soluções completas possíveis (depende do tipo de problema).

Logo, branch and bound difere do backtracking apenas em dois itens durante cada iteração. O valor do bound estabelecido do problema e o valor da melhor solução obtida.