Exercícios - Projeto e Análise de Algoritmos

Estudante: Guilherme Brizzi

1. Descreva as semelhanças e diferentes entre o algoritmo de caminho mais curto e o algoritmo do caixeiro viajante.

Ambos os algoritmos têm a semelhança de buscarem o menor caminho necessário para um deslocamento. Porém, eles diferem no fato de que o algoritmo do caixeiro viajante busca encontrar qual o caminho para uma quantia *n* de nós pelo qual deve passar, enquanto os algoritmos tradicionais de caminho mínimo – como Djikstra e A\* - buscam o menor trajeto de um nó de origem a um outro nó de fim, sem requisito de passagem por outros *n* nós. Dessa forma, o algoritmo do caixeiro viajante tem uma complexidade temporal máxima muito maior, de n! ou 2n, a depender da implementação. Por outro lado, os algoritmos de caminho mais curto tendem a computar seu resultado com complexidade temporal máxima muito menor, de n\*log(n), n2 ou n3.

1. Apresente um exemplo de problema real onde somente a melhor solução é aceita. Também apresente um exemplo de problema real onde uma solução aproximada da melhor também pode ser aceita.

Um problema real onde somente a melhor solução é aceita é o de caminho mínimo, visto que haverá um valor que será a solução ótima. Um exemplo de um problema em que uma solução aproximada também pode ser aceita é do ensalamento em uma universidade – não necessariamente estamos buscando o ensalamento que maximiza a utilização, mas apenas um que aloque salas para todas as aulas.

1. Suponha que estamos comparando soluções do insertion sort e do merge sort no mesmo computador. Para entradas de tamanho n, o insertion sort executa em 8n2 ao quadrado passos e o merge sort em 64nlog2(n). Para quais valores de n o insertion sort é melhor que o merge sort?

Para encontrarmos para quais valores de n o insertion sort tem performance temporal melhor que o merge sort, devemos resolver a seguinte equação:

Usando uma ferramenta de solução, encontramos o valor n aproximado:

1. Qual o menor valor de n, tal que um algoritmo com tempo de execução 100n2 é mais rápido que um algoritmo cujo tempo é 2n, na mesma máquina?

O valor é:

1. Como podemos modificar qualquer algoritmo para termos um bom tempo de execução no melhor caso?
2. Considere o problema de adicionar dois inteiros em binário usando n bits, armazenados em dois arrays A e B. A soma dos dois inteiros deve ser armazenada na sua forma binária em um array C com n + 1 elementos. Implemente esse problema na linguagem de programação que preferir :)