

## Regras para a execução do trabalho:

1. No dia **07/12/2018** às **23:50** encerra-se o prazo para a entrega do trabalho prático. Trabalho atrasados não serão considerados.
2. O trabalho será realizado **individualmente**.
3. O aluno, até o prazo final de entrega, deverá fazer o envio **via Moodle** de: *(i)* arquivos de código-fonte do programa e as instruções de como compilar e/ou executar e *(ii)* relatório sobre a atividade.
4. Códigos mal identados, desorganizados e/ou não comentados não serão considerados (**nota zero**).

## Sobre a avaliação do trabalho:

1. A nota será composta pela qualidade técnica da solução desenvolvida. A nota é condicionada a apresentação em dia/horário previamente determinados.
2. Trabalhos plagiados, com informações parciais e/ou não devidamente apresentados terão notas **zero**.

## Datas:

1. **Apresentação do enunciado do trabalho:** 19/10/2018
2. **Entrega do trabalho:** 07/12/2018 até às 23:50 (via Moodle).

## Definição:

Projete um algoritmo para resolver os problema abaixo utilizando *(i)* força bruta e *(ii)* os paradigmas de projeto de algoritmos (guloso, divisão e conquista ou programação dinâmica). O trabalho consiste em:

1. Projetar soluções para os problemas utilizando *(i)* força bruta *(ii)* um dos paradigmas de projeto de algoritmos (guloso, divisão e conquista ou programação dinâmica).
2. Analisar a complexidade de pior caso dos algoritmos propostos.
3. Implementar os algoritmos propostos e comparar os resultados obtidos quanto a qualidade das soluções e ao tempo de execução.

**Problema A (Elevadores inteligentes).** Eu trabalho em um prédio muito alto com um elevador muito lento. É frustrante quando as pessoas apertam os botões de muitos andares vizinhos (digamos 13, 14 e 15) quando eu vou do térreo para a cobertura. Neste caso, minha viagem até a cobertura é interrompida três vezes, uma vez em cada um desses andares. Seria mais educado se as três pessoas concordassem em pressionar apenas o 14 – nos andares 13 e 15, as pessoas subiriam ou desceriam as escadas em um andar. Sua tarefa é escrever um programa de otimização de elevador. Todas as pessoas dizem de antemão ao entrarem no elevador (no

térreo) a intenção dos andares em que desejam ir. O elevador então decide em quais andares parará. O elevador é limitado a parar em no máximo  $K$  andares de modo a minimizar o número de pessoas que precisam subir/descer escadas. Considere que subir a escada (ou descer) é calculado pela diferença modular entre os andares (isto é, entre o andar desejado e o andar mais próximo em que o elevador parou).

**Problema B (BusAlegrete).** Um sistema de linhas de ônibus pode ser modelado como um grafo direcionado ponderado onde os vértices são os pontos de transferência, as arestas são as rotas que os ônibus tomam entre esses pontos de transferência e os pesos são os tempos que levam para viajar entre os pontos de transferência. Além disso, cada aresta é rotulada com o horário de início e a frequência do serviço de ônibus ao longo dessa rota (se uma aresta é rotulada com (10, 30), então os ônibus saem a cada 30 minutos, começando às 12:10). Modifique o algoritmo de Dijkstra para calcular a maneira mais rápida de se conectar dois pontos do grafo, dada uma determinada hora em que a viagem vai começar.

**Problema C (Tetris com "L").** A empresa de software PampaVision está contratando programadores para desenvolver um novo jogo para auxiliar no desenvolvimento cognitivo dos alunos. O aplicativo a ser desenvolvido é baseado no famoso jogo tipo *Tetris*, porém somente com peças do tipo "L". O jogo deverá atender as seguintes definições. Considere um tabuleiro de tamanho  $n \times n$ , onde  $n$  é um número potência de 2 (e.g., 2, 4, 8, 16...). Uma ou mais posições do tabuleiro, com tamanho  $1 \times 1$  está faltando (não é possível utilizá-la). Peças do tipo "L" são representadas por um quadrado com dimensões  $2 \times 2$  com uma célula de tamanho  $1 \times 1$  em falta/branco. Desenvolva um algoritmo para posicionar o maior número de peças do tipo L no tabuleiro.