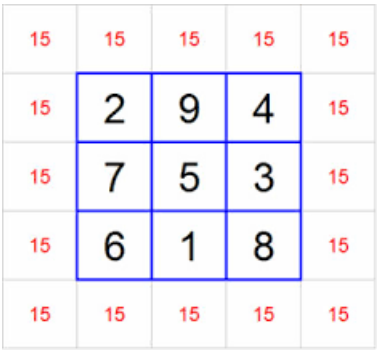
Universidade Federal de Santa Maria
Curso de Ciência da Computação
Disciplina: Computação Gráfica Avançada
Segundo Semestre de 2017
Prof. Cesar Tadeu Pozzer

Trabalho 4 – Busca em espaço de estados



Desenvolva um programa em C/C++ para determinar o número de combinações possíveis em um quadrado mágico. No quadrado mágico a soma dos elementos na vertical, horizontal e diagonais deve ser o mesmo, como mostrado nos exemplos.

O processo de busca é muito semelhante ao usado em jogos de tabuleiro, como xadrez, ou jogos de puzzle, como o Sudoku e Sokoban. Deve-se explorar todas as possibilidades de jogadas.

O programa deve ser genérico a ponto de permitir a definição do tamanho da matriz, que deve ser no mínimo 3x3.

O programa deve imprimir o número de combinações válidas e o tempo necessário para encontrar a solução. Juntamente com o programa deve-se enviar um pequeno relatório com tempos e soluções dos experimentos realizados.

Data e Formato de Entrega:

- Data: 30/out/2017.
- No email e no cabeçalho do arquivo, devem conter o nome completo e matrícula do aluno. O arquivo deve ser enviado para **pozzer3@gmail.com** com subject "**CGA T4**".
- O programa deve ser enviado em um arquivo compactado **fulano.rar** (fulano = login ou nome do aluno), **sem links para dropbox, drive, etc**. Dentro deste arquivo deve haver um diretório com o mesmo nome do arquivo e dentro deste diretório os arquivos do trabalho.
- **Ex:** o arquivo **pozzer.rar** deve conter um diretório chamado pozzer, e dentro do diretório devem estar os arquivos do trabalho. O arquivo .rar pode ser renomeado para .rar.txt caso o gmail não aceite.

Critério de Avaliação:

- **documentação:** descrever no cabeçalho de cada arquivo a ideia geral do código e detalhes específicos de partes que mereçam uma explicação - não comente por exemplo o que faz b++.
- **pontualidade:** Trabalhos não entregues na data não serão avaliados e receberão **nota zero**.
- **legibilidade:** nome de variáveis, estruturação do código.
- **clareza:** facilidade de compreensão - evite códigos complexos e desnecessários. Adote a solução mais simples possível.
- **funcionalidade:** o programa deve satisfazer todos os requisitos. Programas que não compilarem ou que não atenderem a nenhum requisito receberão **nota zero**.

Você pode discutir estratégias e ajudar o colega na implementação, porém evite passar código fonte. Programas semelhantes terão nota zero.