Colégio Técnico de Campinas

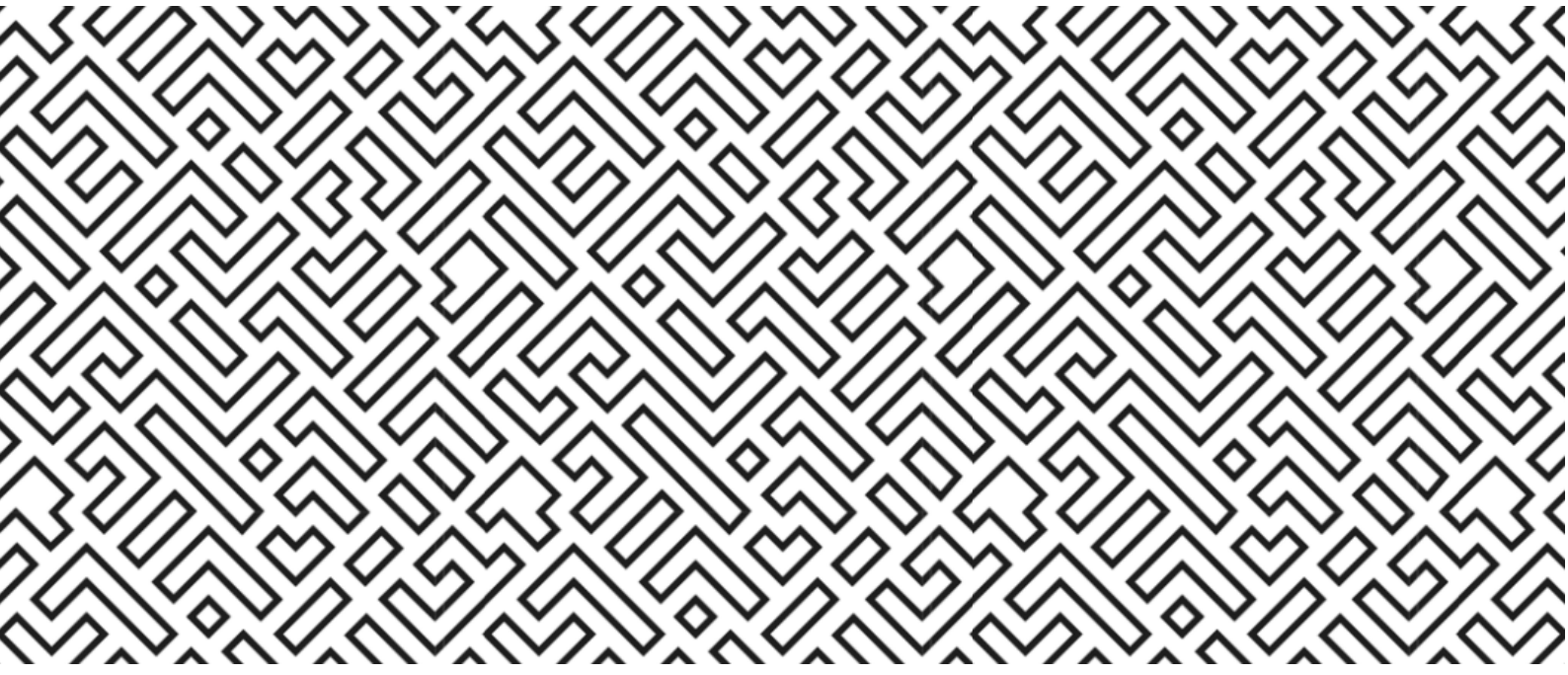
Informática – 4.º semestre

Estrutura de Dados II - TI401

Giovanna Pavani Martelli - 19173

Maria Luiza Sperancin Mancebo - 19186

Relatório do Projeto I: Resolvedor de Labirintos



Campinas – SP

2020

Introdução:

Nos foi proposto a criação de uma aplicação em Windows Forms com C#, usando as classes associadas à PilhaLista que vimos durante as aulas, para ler um arquivo de texto que representa um labirinto e encontrar todas os caminhos possíveis, saindo da posição [1,1] de uma matriz e procurando o caminho até achar uma letra 'S', a saída desse labirinto.

Para isso deveríamos usar a técnica de Backtracking, também vista e lecionada em aula. Fizemos nossa comunicação pelo Discord e compartilhamos a tela para programação pelo TeamViewer.

Desenvolvimento:

21/08/2020 Sexta-Feira:

Antes de iniciar o projeto, lemos novamente toda a parte de Caminho e Backtraking da apostila de Estrutura de Dados II.

Para iniciar, fizemos a parte visual (form), onde colocamos os dois DataGridViews e os dois botões, além das labels indicando para o usuário o que cada DataGridView iria fazer. Em seguida, começamos a classe GrafoBacktraking, onde implementamos o método BuscarCaminhos (todos os caminhos possíveis) e alteramos os seguintes métodos já existentes: Exibir (a matriz do labirinto no DataGridView) e ExibirUmPasso (que desenha no dataGridView um passo). Baseamo-nos na GrafoBacktraking feita pelo nosso professor Chico para fazer essa classe.

Alguns erros ocorreram nela, como por exemplo, usávamos um método ExisteDado na pilha, mas ele não estava funcionando como o esperado, retornando sempre false, o que quebrava o raciocínio do nosso algoritmo.

Além disso criamos a classe Posicao, que possui como atributos uma linha e coluna, para não termos que escrever linha e coluna em todos os lugares, apenas instanciamos uma Posicao (por exemplo, na classe Movimento temos uma Posicao chamada origem e uma direção como atributos, para saber onde que estávamos na matriz e para onde fomos) Ou seja, a criamos apenas por questão de maior organização e alinhamento à orientação a objetos.

28/08 Sexta-Feira:

Fizemos funcionar o botão abrir arquivo (que abre um arquivo .txt, o lê e organiza o labirinto em uma matriz, para depois ser resolvido). Além disso, tentamos resolver o erro do GrafoBacktraking (o mesmo da outra semana

envolvendo o método ExisteDado), porém não conseguimos e guardamos o problema para resolver na semana seguinte. Támbém fizemos o labirinto ser pintado de diferentes formas de acordo com os caracteres que estiverem na matriz (por exemplo, quando tiver # (parede), no DataGridView, será pintado de preto).

04/09 Sexta-Feira:

Criamos o método Copia na classe PilhaLista afim de fazer uma cópia do que for passada por parâmetros, pois estávamos perdendo as informações depois de manipula-las. Também resolvemos que não iriamos mais usar o método ExisteDado, mas, sim colocar caracteres nos lugares que foram empilhados para sabermos se já tínhamos passado pelo local.

Arrumamos alguns erros como a confusão de linha com coluna, falta de ponto e virgula, etc.; além disso, arrumamos o design do Form, afim de ficar mais agradável para o usuário utilizar o aplicativo. Porém, mesmo quase finalizado, os passos não estava sendo exibidos na tela enquanto nosso aplicativo procurava os caminhos, a tela apenas congelava aparecia finalizada segundos depois.

06/09 Domingo:

Para finalizar, conseguimos exibir cada passo do caminho no labirinto durante sua busca, e, após pequenos ajustes e requintes, o projeto finalmente estava completo.

Conclusão:

Usando todo o material fornecido de C# em aula, seguimos as instruções da forma que foi solicitado e, ao término do desenvolvimento, nosso resolvedor de labirinto funcionou como o esperado. Ao realizar esse projeto entendemos melhor como manipular pilhas e como realizar o Backtraking, além de treinar muito nossa lógica de programação e manipulação de telas gráficas em C#.