

INTERFACE HARDWARE E SOFTWARE
PROF. CALEBE CONCEIÇÃO

Atividade Avaliativa - Roteamento

*Realize a atividade que segue. A entrega consiste na submissão no SIGAA de um arquivo ZIP contendo: código-fonte da solução (preferencialmente em Java, Python, C, ou C++) e uma pasta chamada **testes** contendo uma lista de pelo menos 10 grupos de 3 arquivos indexados nomeados no formato **grid_X.txt**, **solucao_X.png**, **resultado_X.txt**. Serão os 5 casos originais fornecidos juntos com o exercício, mais 5 casos gerados por você usando o programa de ajuda disponibilizado. O conteúdo desses arquivos são especificados no corpo desta atividade. Divirtam-se.*

Descrição: Nesta atividade, você está desafiado a implementar um programa que resolva o problema de roteamento em um grid 18x18 para 3 fios de 2 terminais cada. A figura abaixo ilustra um caso de entrada para o problema.

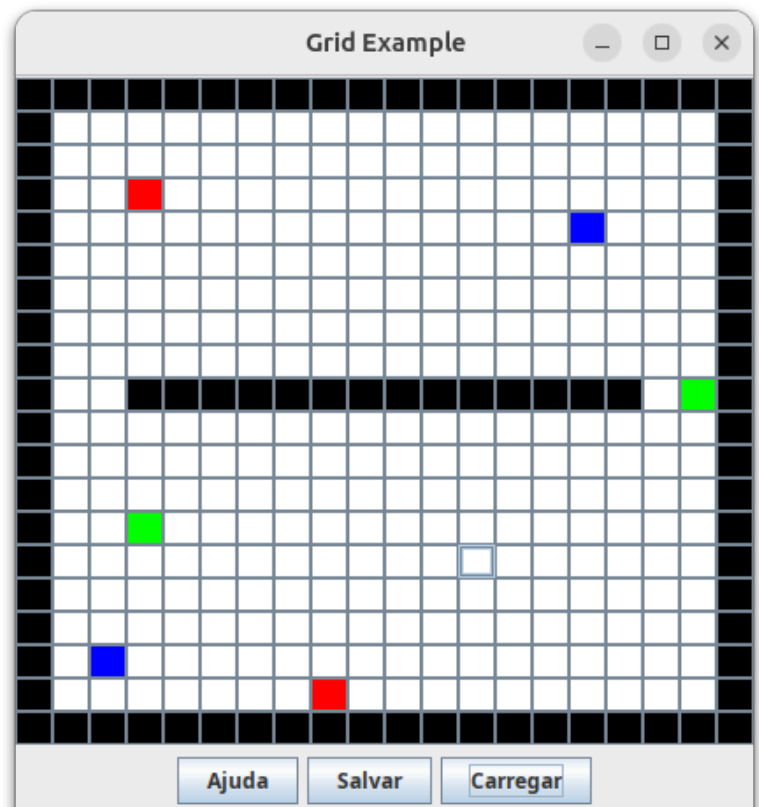
Seu trabalho é desenvolver um programa que leia a configuração do grid conforme a especificação no final desta folha, e apresente como resultado o grid roteado, com ligações entre os pinos de cores iguais. Use a mesma cor para identificar o fio que liga dois pinos.

A entrega consiste no código fonte, e da subpasta contendo 10 trios de arquivos **grid_X.txt**, **solucao_X.png**, **resultado_X.txt** especificados conforme segue.

grid_X.txt → entrada do programa, contem um grid de inteiros de dimensão 20x20, onde 0 (branco) indica espaço livre, 1 (preto) indica barreira, 2 indica um pino vermelho, 3 indica um pino verde, e 4 indica um pino azul.

solucao_X.png → contém a apresentação gráfica do resultado da execução do seu algoritmo para a entrada X, que deve consistir da configuração de entrada acrescida dos caminhos que ligam os pinos vermelho, verde e azul, sem sobreposição (cruzamento), na mesma cor de seus pinos. Observe no verso a saída para o exemplo. Se não houver solução, o programa deve apresentar o roteamento que é possível, e manter sem ligação os pinos que ficaram sem serem roteados. Lembrando que as conexões são sempre nas direções vertical e horizontal, nunca na diagonal (nesse exemplo).

resultado_X.txt → consiste em um arquivo contendo: na primeira linha a indicação de quantos pinos foram roteados (deseja-se que seja sempre 3, mas nem sempre será possível por melhor que seja o algoritmo); a segunda contém o somatório do comprimento das ligações realizadas, e a partir da terceira linha o grid de inteiros que

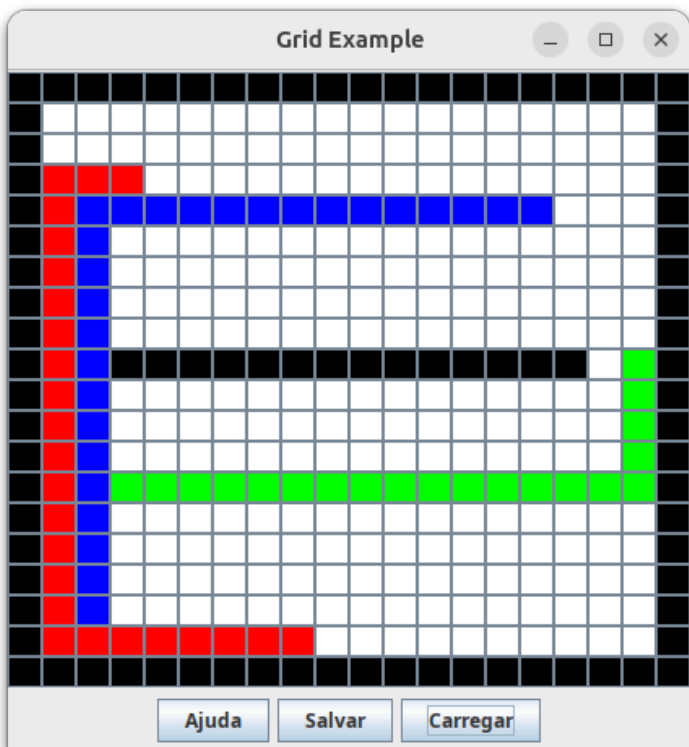


corresponde à solução para aquela entrada X, com dimensão 20x20.

Para esta atividade, estou preocupado em algoritmos que encontrem soluções, não precisa ser o mais eficiente. Mas, claro, quero poder comparar as diferentes soluções entregues. Por isso escolhi a métrica do somatório dos comprimentos das ligações (incluindo os pinos), que você deve reportar no arquivo de saída de cada execução.

Use o programa em Java fornecido junto com este exercício para gerar mais cenários de entrada para testar e para apresentar o resultado gráfico do seu algoritmo (carregando o arquivo grid gerado). Use uma ferramenta de captura de tela do seu computador para salvar a imagem do resultado (*quero evitar a fadiga de implementar o botão de exportar imagem na aplicação*). Estou disponibilizando o fonte da aplicação de ajuda pra quem se estiver afim de aumentar o grid, a quantidade de pinos, ou mesmo implementar mais funcionalidades... Por exemplo, sugiro implementar um debug, para visualizar a execução do algoritmo passo a passo. Ideias tenho, falta-me tempo! :-)

*Uma solução possível para o exemplo de entrada
(conteúdo do arquivo PNG)*



*Uma solução possível para o exemplo de entrada
(conteúdo do arquivo resultado)*

```
3
72
11111111111111111111
10000000000000000001
10000000000000000001
12220000000000000001
12444444444444444001
12400000000000000001
12400000000000000001
12400000000000000001
12400000000000000001
12411111111111111031
124000000000000000031
124000000000000000031
124000000000000000031
12433333333333333331
12400000000000000001
12400000000000000001
12400000000000000001
12400000000000000001
12222222200000000001
11111111111111111111
```

O mais importante de tudo: Divirtam-se fazendo! 😊