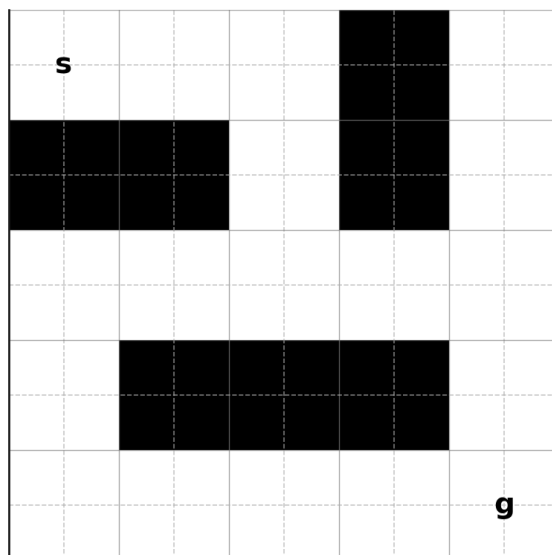


PCC104 – Prova 3

Universidade Federal de Ouro Preto

Nome do(a) aluno(a): _____

1. (2 pt) Apresente a análise de complexidade de *pior caso*, em número de comparações, da sua implementação do algoritmo *Selection Sort*.
2. (2 pt) Considere a sua implementação de *heap*. Descreva o comportamento do programa quando o usuário tenta inserir uma chave que já está presente na estrutura. Justifique citando os trechos de código relevantes.
3. (2 pt) Apresente a análise de complexidade do método **search** da sua Árvore Binária nos seguintes cenários:
 - (a) árvore perfeitamente balanceada;
 - (b) árvore degenerada (todos os nós à direita ou à esquerda).
4. (2 pt) Numere as células do labirinto abaixo seguindo a ordem em que são removidas da fronteira (*frontier*) pela sua implementação do algoritmo de Busca em Largura (*Breadth-First Search*). Comece do vértice de origem, **s**. O objetivo é chegar até **g**.



5. (2 pts) Considere um conjunto de n turmas, cada uma com um número de estudantes s_i , e um conjunto de m salas, cada uma com capacidade c_j ($1 \leq i \leq n$, $1 \leq j \leq m$). Deseja-se atribuir cada turma a exatamente uma sala de forma que $s_i \leq c_{(\text{sala atribuída})}$. Caso não exista atribuição viável para todas as turmas, o algoritmo deve sinalizar falha. Escreva um algoritmo em *pseudocódigo* utilizando **backtracking** que encontre uma alocação válida (ou indique que ela não existe). Você pode definir métodos auxiliares se achar necessário.