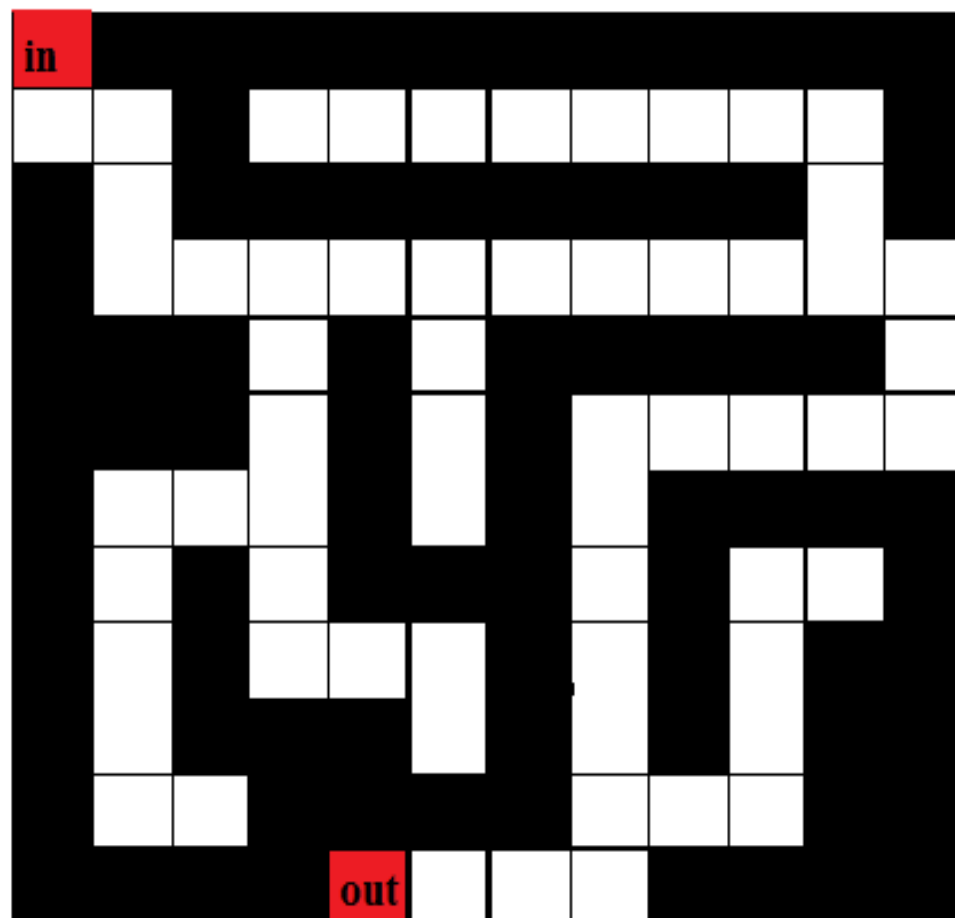


# Aplicação de pilhas – Problema do labirinto

- Adaptado de Horowitz, E. & Sahni, S. “Fundamentos de Estruturas de Dados”. Ed. Campus. 1984.
- Simulação do experimento da curva de aprendizado;
- Experimento real: avalia o tempo que uma cobaia (rato) leva para aprender o caminho de saída de um labirinto.

Labirinto



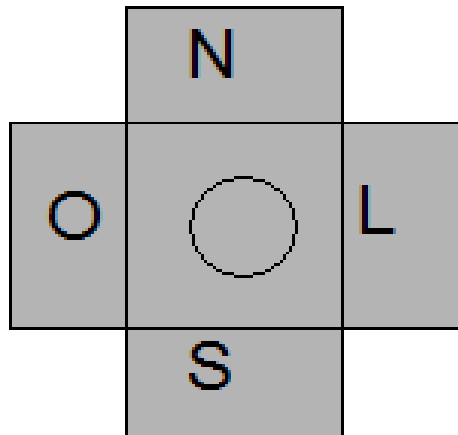
Representado pela matriz Maze

0											
0	0		0	0	0	0	0	0	0	0	0
	0										0
	0	0	0	0	0	0	0	0	0	0	0
			0		0						0
			0		0		0	0	0	0	0
	0	0	0		0		0				
	0		0				0		0	0	
	0		0	0	0		0		0		
	0				0		0		0		
	0	0					0	0	0		
				0	0	0	0				

N linhas X M colunas

- Nessa versão o rato se move como a torre no xadrez porém restrito a uma “célula”
- A direção do próximo movimento do rato (Norte, Sul, Leste ou Oeste) é determinada pela análise de uma vizinhança ao redor da posição atual;
- O rato se movimenta de célula em célula, ele não salta;
- As células do caminho são vizinhas.

- A direção do movimento do rato (Norte, Sul, Leste ou Oeste) é determinada pela análise de uma vizinhança ao redor da posição atual;
- A determinação da direção pode ser randômica



Uma tabela representa os incrementos necessários ao deslocamento do rato na direção determinada (N,S, L ou O);

Ex: Posição atual é (i,j)

Movimento para Leste

Nova posição=(i,j)+DIREC(Leste)=(i+0,j+1)

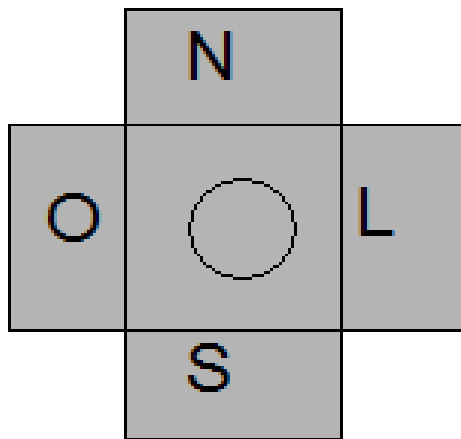
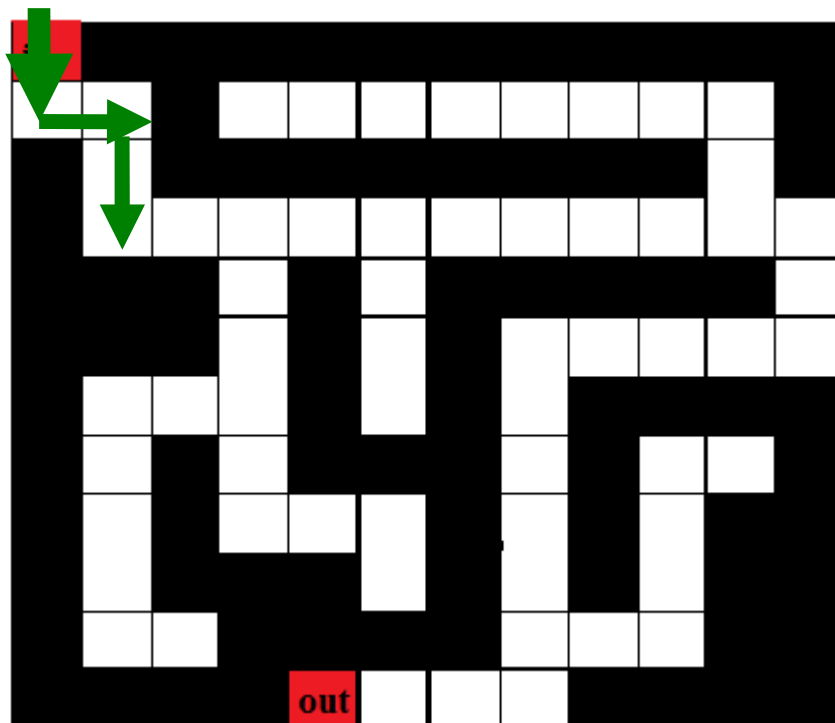


Tabela Direcionadora

	Incremento linha	incremento coluna
Norte	-1	0
Sul	+1	0
Leste	0	+1
Oeste	0	-1

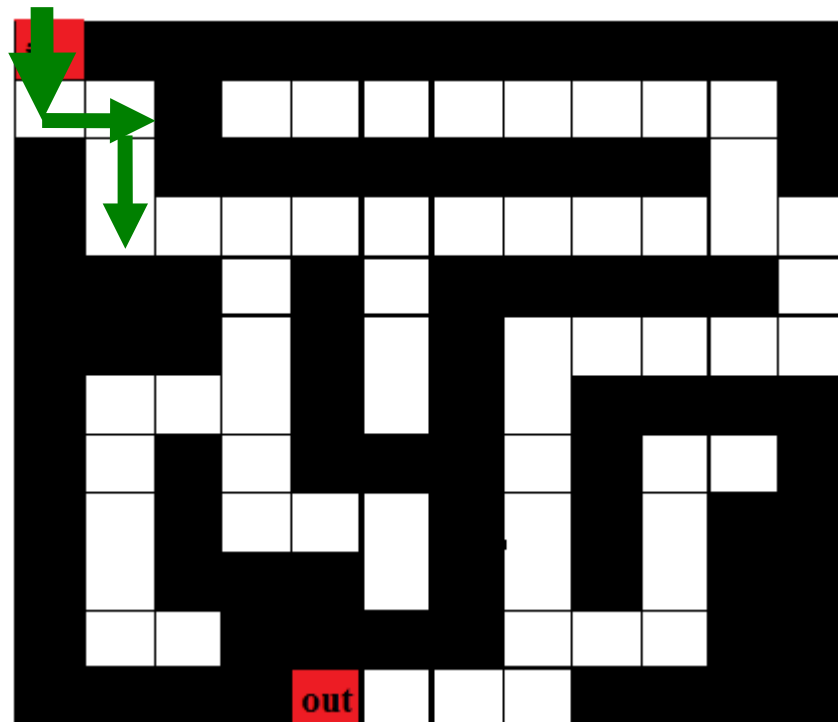
- Uma matriz auxiliar (MARK) é utilizada para marcar as células já visitadas de maneira que o rato não repita caminhos já explorados



MARK

+											
+	+		0	0	0	0	0	0	0	0	0
	+										0
	+	0	0	0	0	0	0	0	0	0	0
			0		0						0
			0		0		0	0	0	0	0
	0	0	0		0		0				
	0		0				0		0	0	
	0		0	0	0		0		0		
	0				0		0		0		
	0	0					0	0	0		
				0	0	0	0				

- Cada célula visitada é empilhada juntamente com a direção tomada para alcançá-la;



**PILHA**

**TOPO** → (3,2,Sul)  
(2,1,Leste)  
(2,1,sul)  
(1,1,sul)

- Dados a serem empilhados

Linha;

Coluna;

Direção que levou a (linha,coluna);

```
typedef struct{  
    int lin;  
    int col;  
    int mov;  
}.....
```



caminho(MAZE,DIREC, k,l) // (k,l) coords. da saída

Declara MARK,M,N; Cria PILHA;

Empilha(1,1,2); // 1,1 é o início do labirinto

ENQUANTO(PILHA Ñ VAZIA)

(i,j,mov) = desempilha;

mov=mov+1; // ou rand()

ENQUANTO (1 <= mov <= 4)

g=i+DIREC(mov,1); h=j+DIREC(mov,2);

SE(g==k E h==l) // encontrou a saída

ExibeCaminho(Pilha,g,h,i,j); RETURN;

SENÃO

SE (MAZE(g,h)==0 E MARK(g,h)==0)

MARK(g,h)=1;

empilha(i,j,mov);

mov=0; g=i,h=j;

mov=mov+1;

NÃO encontrou a saída

ExibeCaminho(Pilha,g,h,i,j)

display(g,h);

display(i,j)

ENQUANTO(PILHA Ñ VAZIA)

(a,b)=desempilha;

display(a,b);