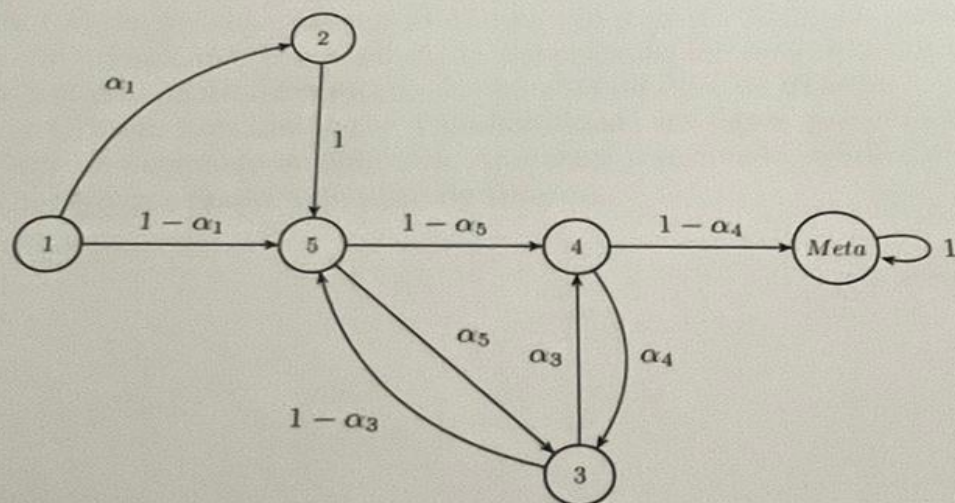


8.0 **1)** Considere um jogo de tabuleiro para crianças muito simples e que envolve 2 jogadores de cada vez. Neste jogo, cada jogador parte da casa 1 e ganha o que chegar mais depressa à casa *Meta*.

O percurso de cada jogador desde a casa 1 até à casa *Meta* segue o diagrama a seguir, que indica as casas para onde o jogador se pode mover durante o jogo. A escolha do percurso depende do resultado do lançamento de uma moeda ao ar que não é equilibrada (moeda diferente em cada casa). O resultado do lançamento em cada casa depende de um parâmetro α_i , sendo i o número da casa. Nas alíneas seguintes assumo $\alpha = [0.2, 0.3, 0.1, 0.45]$.



2.5 **1.a)** Represente em Matlab a matriz de transição T , em que T_{ji} representa a probabilidade de o estado j se seguir ao estado i .

Código Matlab:

- 2.5 **1.b)** Considere o percurso mais pequeno, isto é, o trajeto que passa pelo menor número de casas. Determine a probabilidade do jogador seguir esse percurso para chegar à Meta.

Resposta: _____

Código Matlab: _____

- 3.0 **1.c)** Determine o número médio de casas percorridas desde o início até ao fim do jogo (incluindo a casa inicial e a casa final).

Resposta: _____

Código Matlab: _____

- 6.0 **2)** Considere a implementação de um filtro de Bloom para representar os elementos de um conjunto de palavras. Considere palavras geradas aleatoriamente com caracteres de 'a' a 'z' (equiprováveis) e com comprimento de 5 caracteres (probabilidade de 40%) ou 8 caracteres (probabilidade de 60%).
- 2.0 **2.a)** Assuma que vai adicionar 300 palavras ao filtro de Bloom. Assuma também que o filtro de Bloom usa uma única função de dispersão e que deverá ter uma taxa de falsos positivos de aproximadamente 3%. Determine o tamanho adequado do filtro de Bloom e apresente o código que usou.

Resultado: _____

Código Matlab:

- 4.0 **2.b)** Inicialize o filtro de Bloom usando o tamanho calculado anteriormente (se não o conseguiu calcular utilize o valor de 10000). De seguida inclua no filtro 300 palavras diferentes geradas aleatoriamente com as características definidas acima. Use a função de dispersão *default* providenciada pela função *string2hash* (usada nas aulas práticas) na implementação das funções do filtro de Bloom.
- Finalmente, determine por simulação a probabilidade de falsos positivos do filtro implementado, usando um conjunto de adequado de palavras. Apresente o resultado obtido e o código Matlab que usou (não apresente o código das funções do filtro de Bloom).

Resultado: _____

Código Matlab:

6.0 3) Considere as 4 frases seguintes.

1. Lisboa e Porto são duas cidades importantes de Portugal.
2. Lisboa é a capital de Portugal.
3. Porto é a capital do Norte de Portugal
4. Portugal não é só Lisboa e Porto.

Implementando MinHash, determine as assinaturas de cada frase considerando todos os seus caracteres (incluindo ponto final e espaços), estime a distância de Jaccard entre todas as frases e apresente o par mais similar entre todas. Na implementação tem de usar a função de dispersão `hfl()` disponível em <https://bit.ly/3LaAynd>, aplicar 4 funções de dispersão; usar shingles de comprimento $k=3$.

Resposta

Assinatura: Frase 1 = _____
Frase 2 = _____
Frase 3 = _____
Frase 4 = _____

Distância: 1 vs 2 = _____
2 vs 3 = _____

Most similar: 1 vs 3 = _____
2 vs 4 = _____
1 vs 4 = _____
3 vs 4 = _____

Código Matlab (implementação de MinHash e calc. distância):