Exercícios tentativa e erro

Implemente e compartilhe com seus colegas métodos/funções que resolvam, utilizando tentativa e erro, os seguintes problemas:

1. **Mochila Binária (problema de maximização):** dado um conjunto de objetos que possuem dois atributos: valor e peso e a capacidade de uma mochila (peso que ela comporta), desenvolva um método que retorno o valor máximo possível que pode ser colocado dentro da mochila.
   Dica: o controle do status atual do algoritmo pode ser o índice do objeto atual (cada chamada recursiva irá para o objeto seguinte ao atual). Há duas ações possíveis: colocar ou não o objeto atual na mochila. Uma ação possível será plausível se o objeto atual couber no espaço disponível na mochila.

2. **Coloração de Mapas 1 (problema para encontrar uma solução):** dado um mapa representado como um arranjo de países (e uma representação dos vizinhos de cada país [como matriz de adjacência ou lista de adjacências]) e um número inteiro de cores, implemente um método que verifique se é possível colorir todos os países com esse número de cores de forma que dois países vizinhos sejam coloridos com cores diferentes.
   Dica: o controle do status atual do algoritmo pode ser o índice do país atual (cada chamada recursiva irá para o país seguinte ao atual). O número de ações possíveis corresponde ao número de cores. Uma ação possível será plausível se a cor atual não tiver sido usada em nenhum dos países vizinhos ao país atual.

3. **Coloração de Mapas 1 (problema de minimização):** dado um mapa representado como um arranjo de países (e uma representação dos vizinhos de cada país [como matriz de adjacência ou lista de adjacências]) implemente um método que retorne o menor número de cores necessárias para colori os países desse mapa de forma que dois países vizinhos sejam coloridos com cores diferentes.
   Dica: o controle do status atual do algoritmo pode ser o índice do país atual (cada chamada recursiva irá para o país seguinte ao atual). O número de ações possíveis corresponde ao número máximo de cores já usadas mais uma. Uma ação possível será plausível se a cor atual não tiver sido usada em nenhum dos países vizinhos ao país atual.

4. **Ciclo Hamiltoniano (problema de encontrar uma solução**): dado um conjunto de cidades (e uma representação dos vizinhos de cada cidade [como matriz de adjacência ou lista de adjacências]) verifique se existe um ciclo hamiltoniano, isto é, se é possível sair de uma cidade e retornar a ela, passando por todas as outras cidades uma única vez.
   Dica: o algoritmo pode usar um arranjo do tamanho do número de cidades que conterá o ciclo que está sendo construído; o controle do status atual do algoritmo pode ser o índice neste arranjo (cada chamada recursiva irá tentar preencher a próxima posição do arranjo). Você pode colocar qualquer país na posição inicial. O número de ações possíveis corresponde ao número de países vizinhos ao país que foi colocado na posição anterior desse arranjo. Uma ação possível será plausível se o país atual ainda não tiver sido usado no caminho atual.

5. **Caixeiro Viajante (problema de minimização):** dado um conjunto de cidades (e uma representação dos vizinhos de cada cidade [como matriz de distâncias ou lista de adjacências com pesos/distância]) identifique qual é a menor distância percorrida de forma a se sair da cidade de origem e se retornar a ela, passando uma única vez em cada uma das outras cidades.
   Dica: o algoritmo pode usar um arranjo do tamanho do número de cidades que conterá o ciclo que está sendo construído; o controle do status atual do algoritmo pode ser o índice neste arranjo (cada chamada recursiva irá tentar preencher a próxima posição do arranjo). Você pode colocar qualquer país na posição inicial. O número de ações possíveis corresponde ao número de países vizinhos ao país que foi colocado na posição anterior desse arranjo. Uma ação possível será plausível se o país atual ainda não tiver sido usado no caminho atual e se o custo atual do caminho for menor do que o melhor custo já encontrado (este último critério é apenas de otimização).