sábado, 11 de Março de 2017

Lista adjacências

Resolução com grafo dirigido

Inicialização e deteção de alguns casos não possíveis

- Criação de array com as cabeças das listas e de um array de V tamanho chamado vetor com todos os elementos a 0 (O = V + 1)
- Criação de listas de adjacências (O = A)
  - quando a lista de adjacências é criada sem feitas apões para mais tarde saber qual elemento sem entradas (**A),** isto é feito colocando um 1 a sempre que o vertice aparece a direita num par
- Descobrir qual o primeiro vertice, basta descobrir qual a posição no vetor que esta a 0 (**A**) (O = V)
- Descobrir qual o ultimo elemento, basta verificar se o elemento a seguir a todas as cabeças é ele próprio (**B**) (O = V)

Core do programa e deteção dos restantes casos não possíveis

- Através de uma função usada recursivamente, é procurado o caminho mais longo e são eliminados todos arcos para trás e com deteção de ciclos
- O problema deste é que usa uma função recursiva e no pior caso percorre todas as arestas duas vezes o que significa que e 2^n

ATENÇÃO: um vertice só e o seguinte caso só tenha 1 entrada e só e anterior caso só tenha uma saída

NOTA A: O elemento sem entradas significa que é o primeiro elemento, caso haja mais que um significa que é insuficiente, caso não haja nenhum significa que é incoerente porque significa que há um ciclo

NOTA B: O elemento sem saídas significa que é o ultimo elemento, caso haja mais que um significa que é insuficiente, caso não haja nenhum significa que é incoerente porque significa que há um ciclo

ERROS INPUT: um par apontar para si próprio, o numero ser <1 ou >V