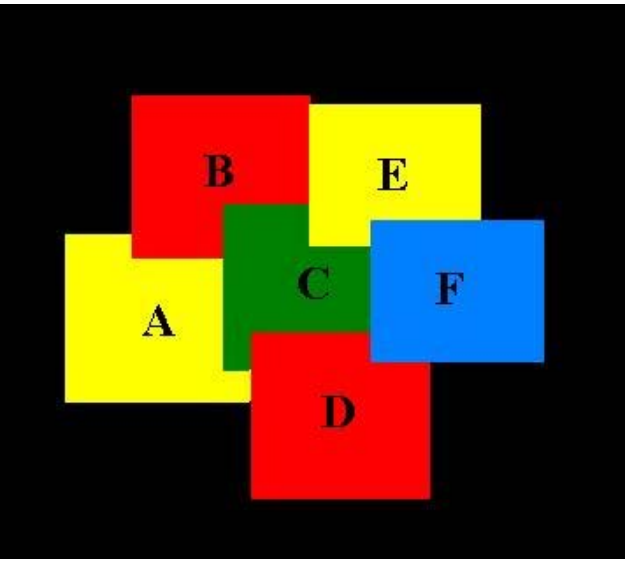
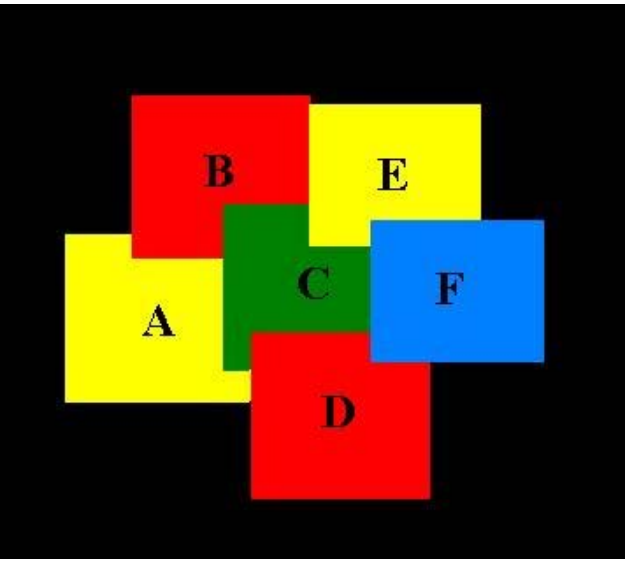
Grafos

Coloração



Lista de Adjacências para a região A: [B, C, D]
Lista de Adjacências para a região B: [A, C, E]
Lista de Adjacências para a região C: [A, B, D, E, F]
Lista de Adjacências para a região D: [A, C, F]
Lista de Adjacências para a região E: [B, C, F]
Lista de Adjacências para a região F: [C, D, E]



O procedimento para se atribuir as cores certas a cada região é o seguinte:

1. Escolhe-se uma região inicial, como por exemplo a região A e atribui-se uma cor a ela.
2. Para atribuir uma cor para B é verificado se dentre as cores existentes, existe uma que não esteja colorindo nenhuma região adjacente a B, então essa cor deverá ser escolhida. Se todas as cores existentes estiverem sendo utilizadas em regiões vizinhas a B, então uma nova cor é criada.
3. O raciocínio é repetido analogamente para cada uma das regiões subsequentes.

Número cromático X(G)

- O **número cromático X(G)** de um grafo *G* é o menor número de cores *k* tal que existe uma *k*-coloração para *G*.
- Para grafos planares, o problema de coloração está intimamente ligado ao problema das 4 cores em mapas.

Coloração – busca em profundidade

Programa Principal

montar a lista de adjacências

inicializar a estrutura de cores

escolher o vértice *Vi* de maior grau para ser colorido primeiro

chamar a sub-rotina Colore_Vertice para colorir o vértice *Vi* escolhido

Sub-rotina Colore_Vertice: *Vk*

se o vértice *Vk* ainda não foi colorido

procurar a cor *C* apropriada

se não existir cor apropriada para colorir o vértice *Vk*

criar uma nova cor *C*

fim se

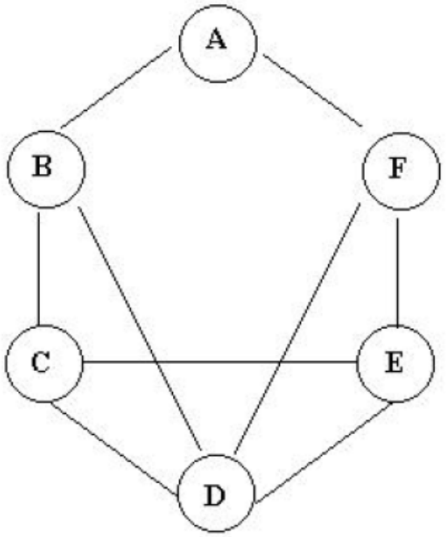
colorir o vértice *Vk* com a cor *C*

para todo vértice *Vj* adjacente a *Vk* faça

chamar a sub-rotina Colore_Vertice para colorir o vértice *Vj*

fim se

Coloração – Busca em profundidade



Grafo

Lista de Adjacência

Vértice A → B → F

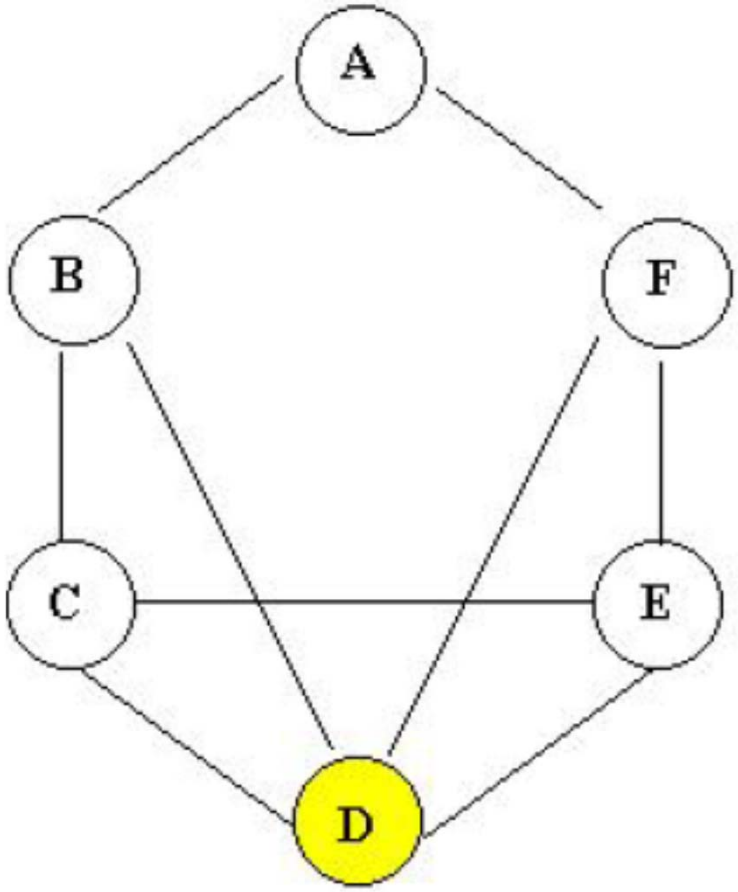
Vértice B → A → C → D

Vértice C → B → D → E

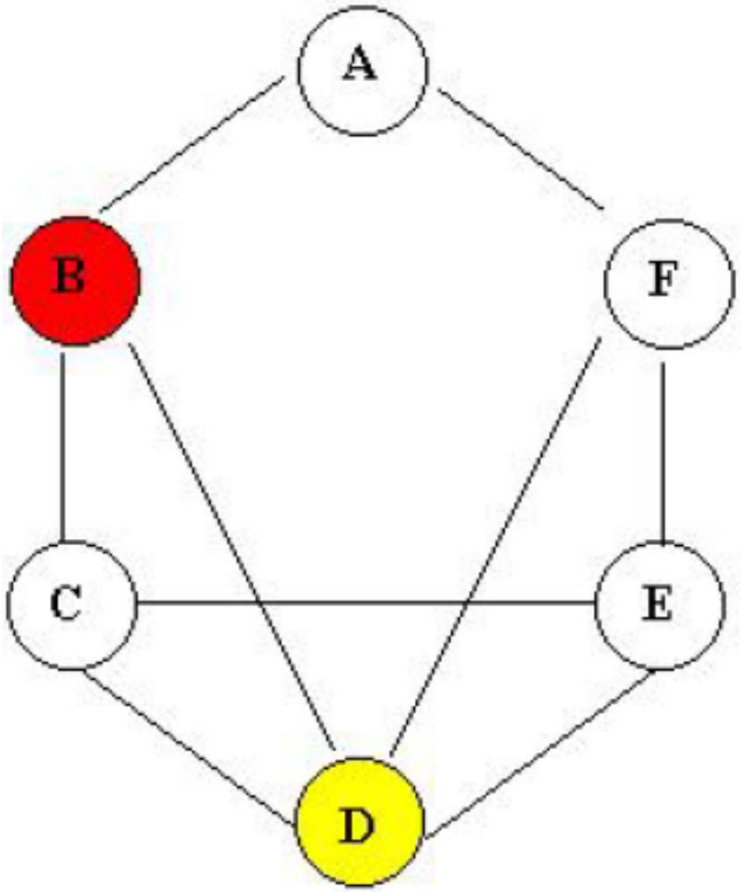
Vértice D → B → C → E → F

Vértice E → C → D → F

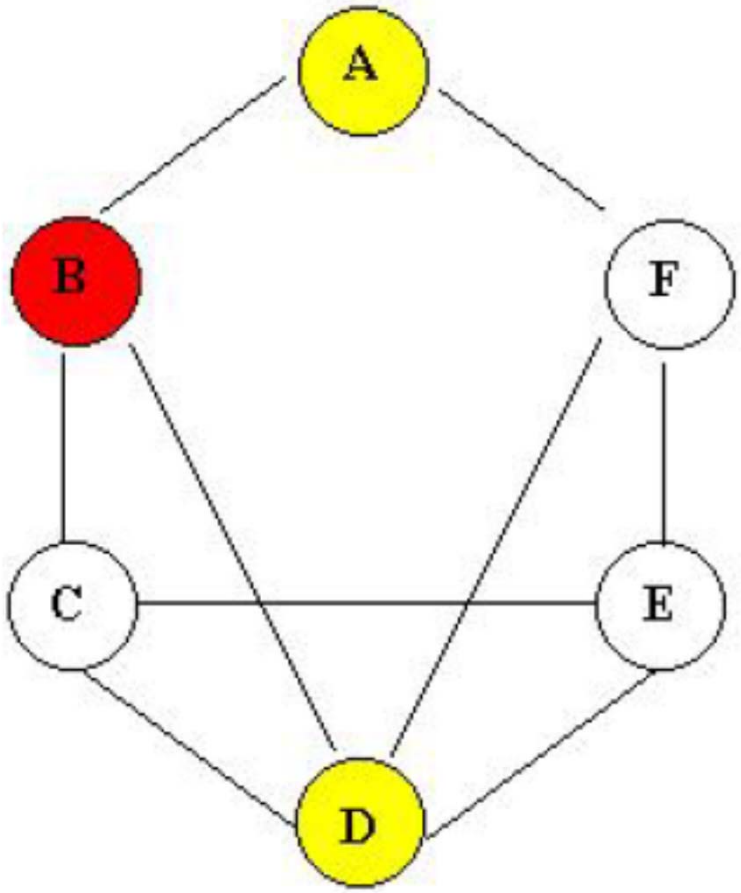
Vértice F → A → D → E



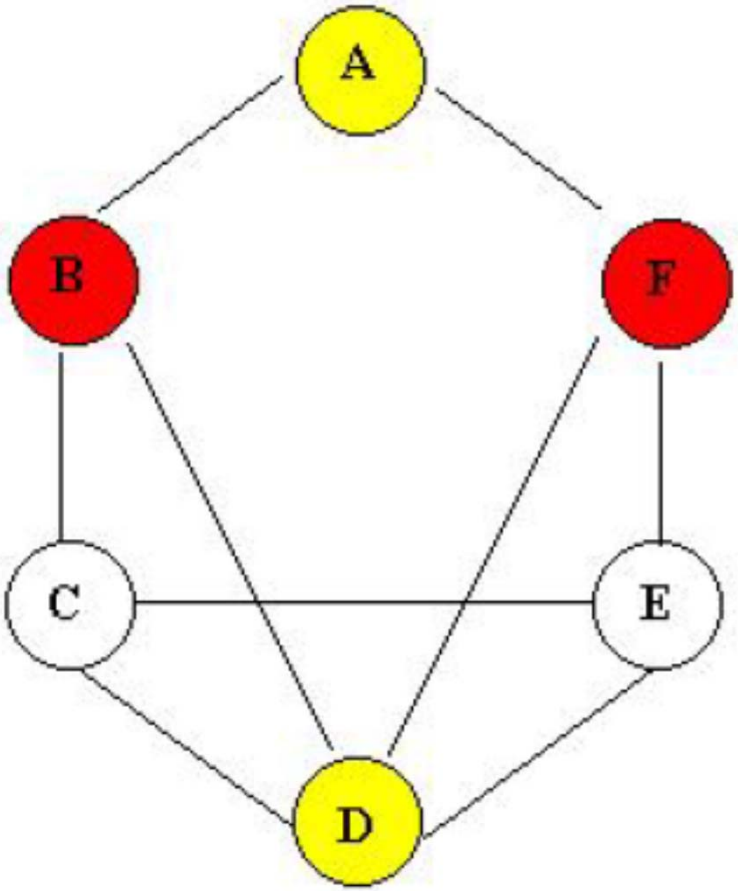
Passo 1



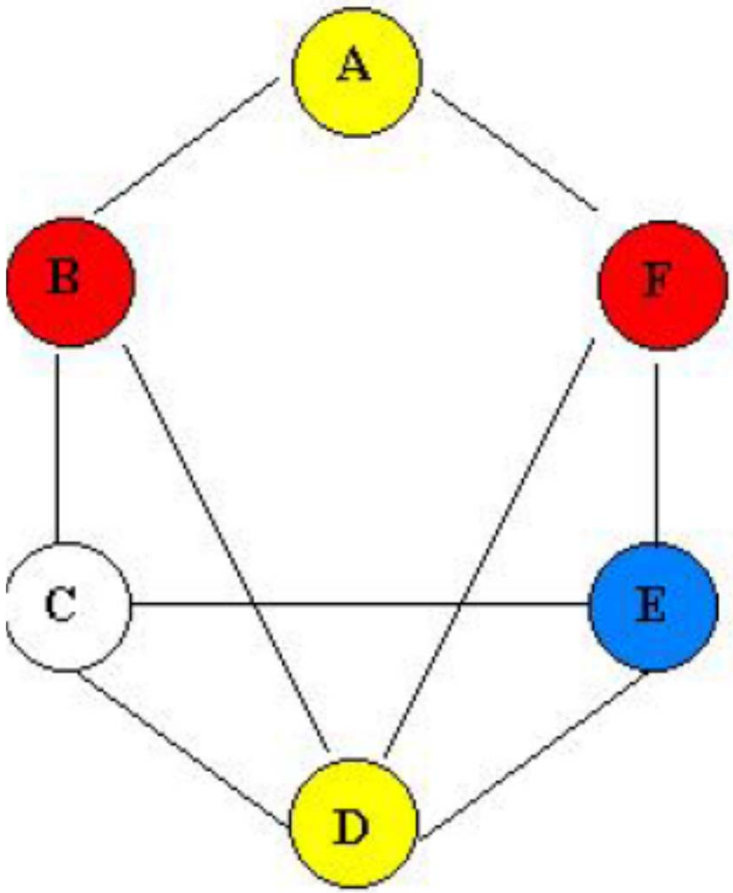
Passo 2



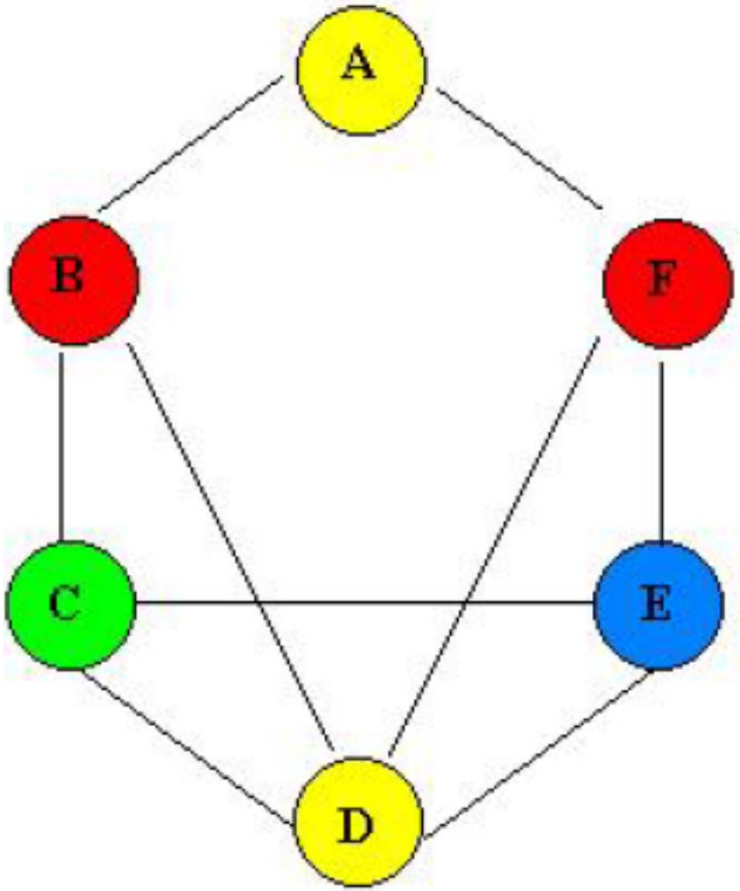
Passo 3



Passo 4



Passo 5



Passo 6

Custo computacional

Considerando que N seja o número de vértices, E o número de arestas, NC seja o número de cores e NVZ seja o número de vizinhos(vértices adjacentes), temos:

1. montar a lista de adjacência= O(N+E)
2. escolher o vértice de maior grau= O(N)
3. procurar cor= O(NVZ*NC)
4. colorir o vértice= O(1)
5. para todos os vértices colorir todos os seus adjacentes: O((N1)*(NVZ*NC))

- Custo Total= O(N+E) + O(N) + O(N*(NVZ*NC)),

de onde podemos concluir que o custo total é da ordem de O(N*(NVZ*NC)), em que no pior caso teremos um custo de ordem cúbica.

Coloração – busca em largura

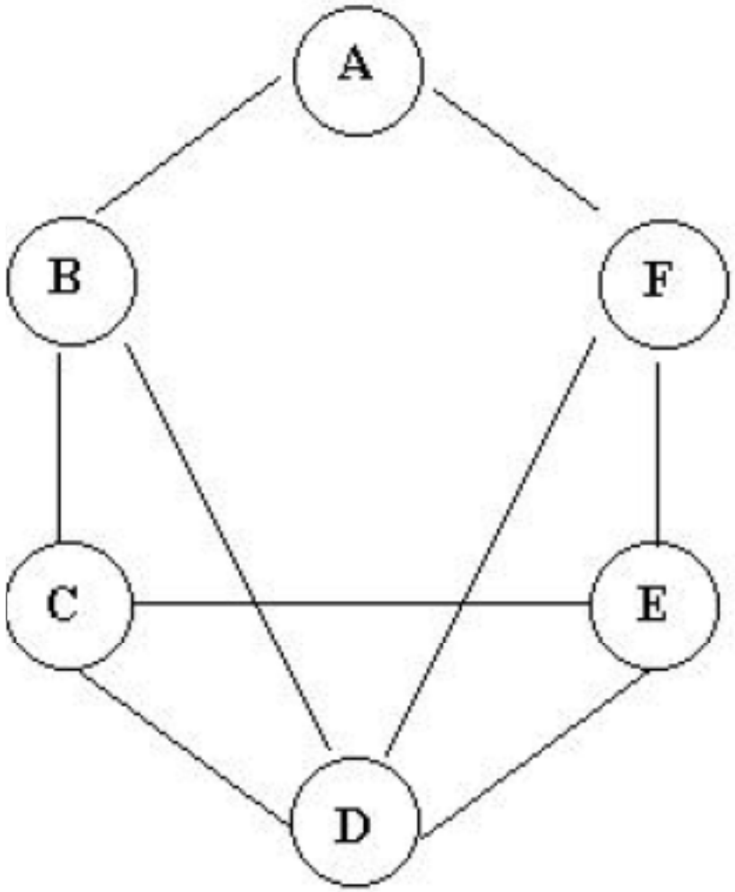
Programa Principal

montar a lista de adjacências
inicializar a estrutura de cores
inicializar a estrutura de fila
escolher o vértice *Vi* de maior grau para ser colorido primeiro
chamar a sub-rotina Colore_Vertice para colorir o vértice *Vi* escolhido
inserir o vértice *Vi* na fila Q
enquanto a fila Q não estiver vazia faca
    remove o vértice *Vk* da fila
    para todo vértice *Vj* adjacente a *Vk* faça
        chamar a sub-rotina Colore_Vertice para colorir o vértice *Vj*
        inserir *Vj* na fila
    fim para
fim enquanto

Sub-rotina Colore_Vertice: *Vk*

se o vértice *Vk* ainda não foi colorido
    procurar a cor *C* apropriada
    se não existir cor apropriada para colorir o vértice *Vk*
        criar uma nova cor *C*
    fim se
    colorir o vértice *Vk* com a cor *C*
fim se

Coloração – busca em largura



Grafo

Lista de Adjacência

Vértice A → B → F

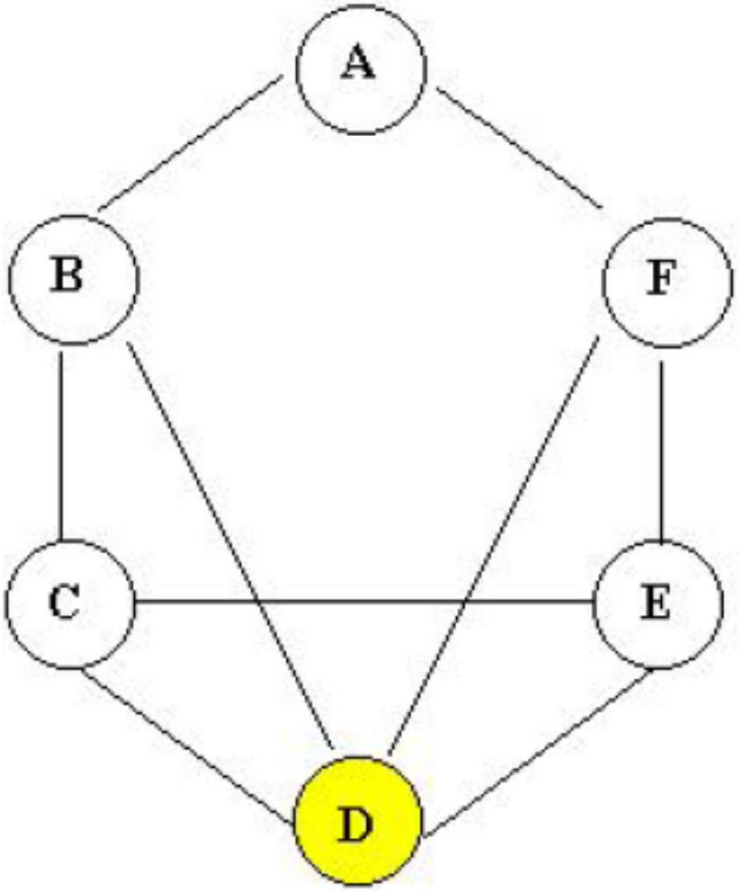
Vértice B → A → C → D

Vértice C → B → D → E

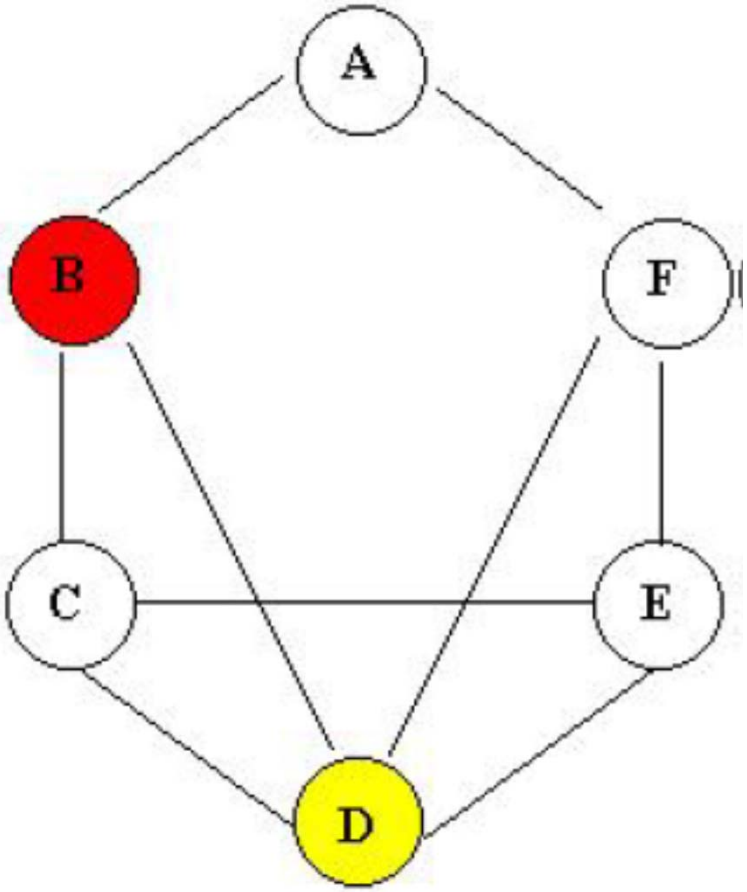
Vértice D → B → C → E → F

Vértice E → C → D → F

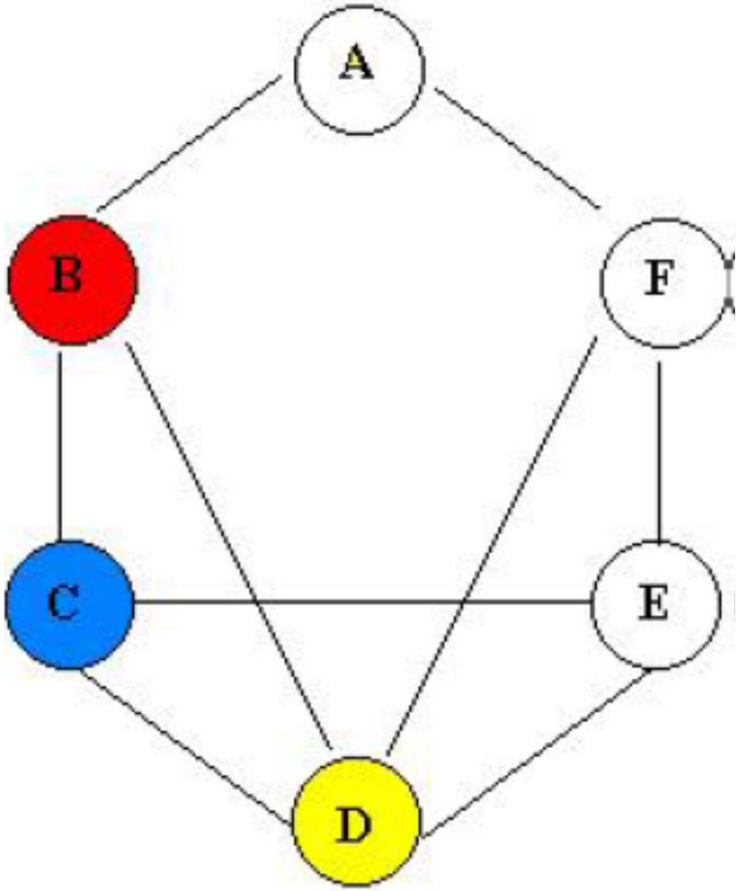
Vértice F → A → D → E



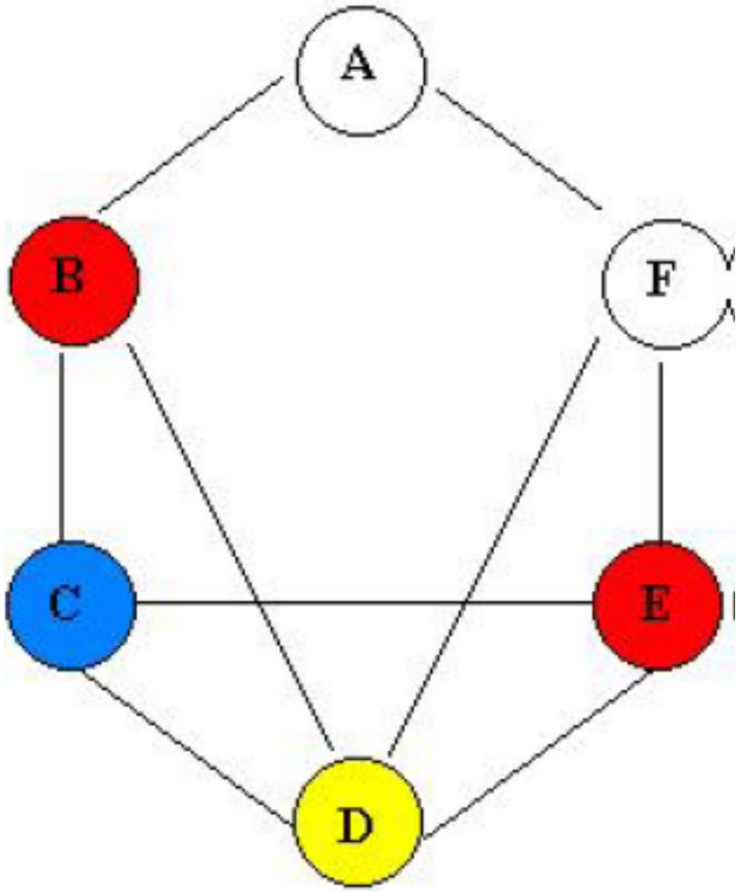
Passo 1



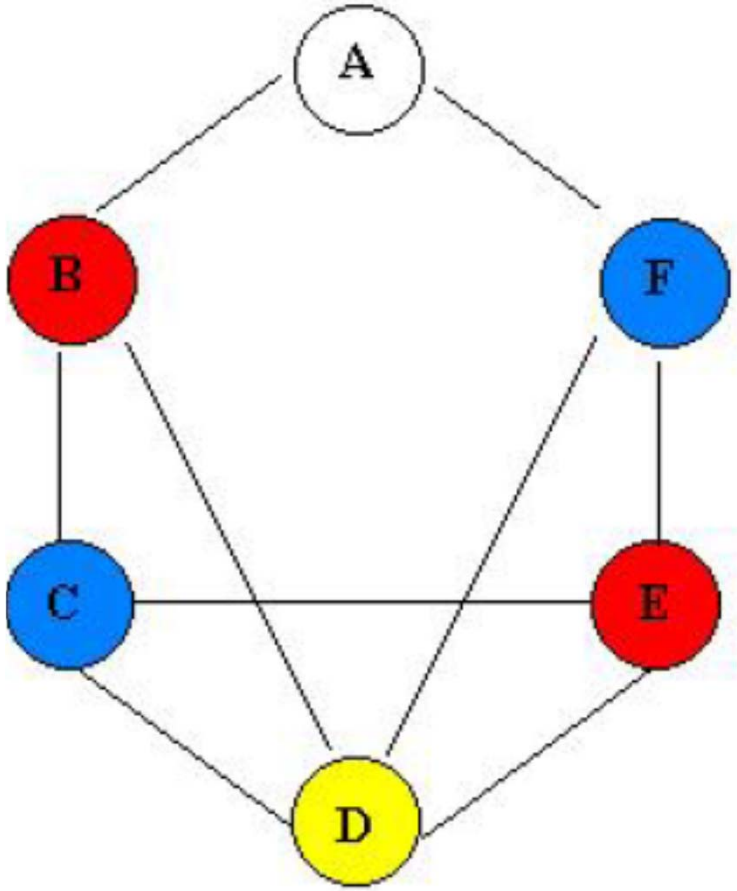
Passo 2



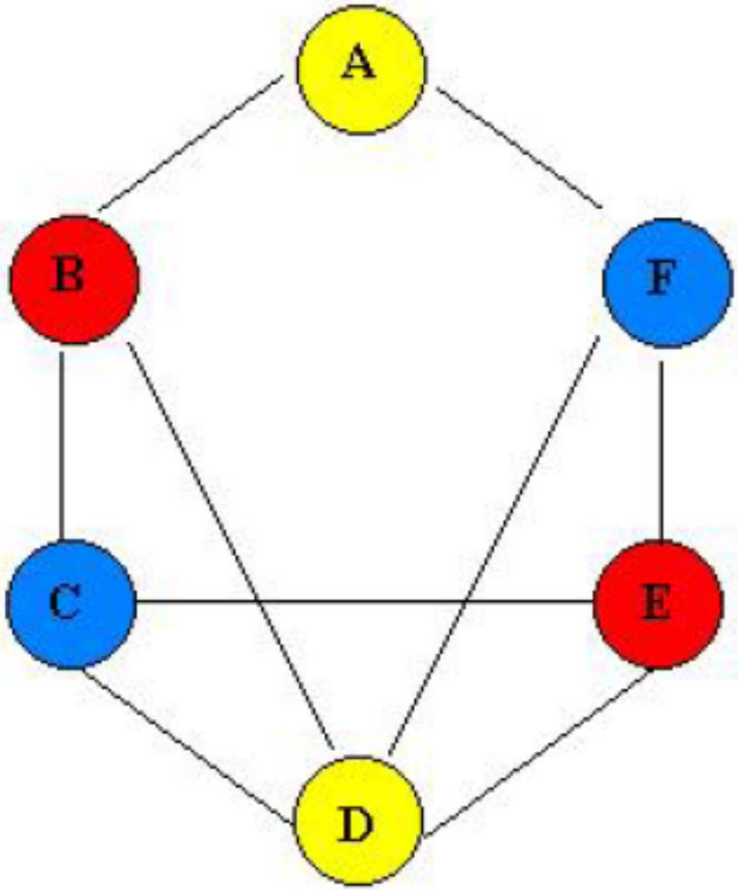
Passo 3



Passo 4



Passo 5



Passo 6

Custo computacional – coloração – busca em largura

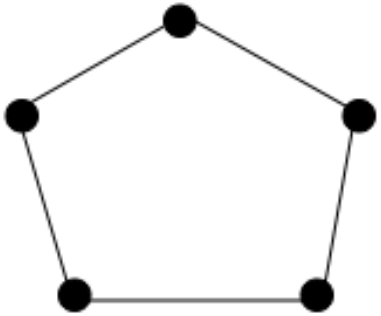
Considerando que N seja o número de vértices, E o número de arestas, NC seja o número de cores e NVZ seja o número de vizinhos(vértices adjacentes), temos:

1. montar a lista de adjacência= O(N+E)
2. escolher o vértice de maior grau= O(N)
3. procurar cor= O(NVZ*NC)
4. colorir o vértice= O(1)
5. inserir e remover da fila= O(1)
6. para todos os vértices colorir todos os seus adjacentes: O((N1)*(NVZ*NC))

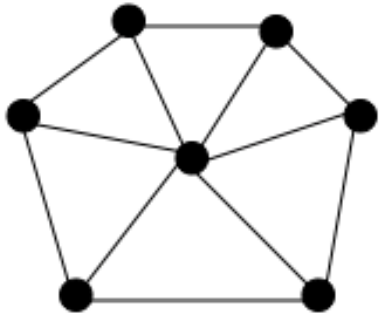
- Custo Total= O(N+E) + O(N) + O(N*(NVZ*NC)),

de onde podemos concluir que o custo total é da ordem de O(N*(NVZ*NC)), em que no pior caso teremos um custo de ordem cúbica.

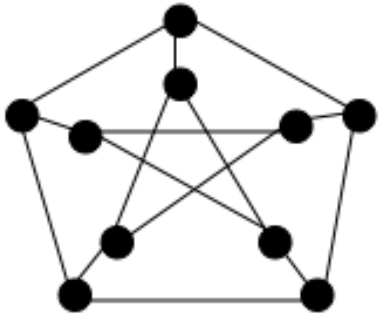
Exercícios



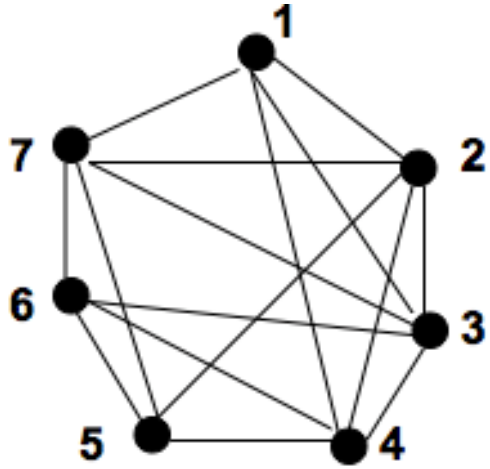
C_n
com n ímpar



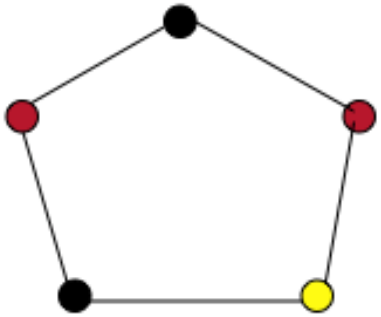
W_n
com n par



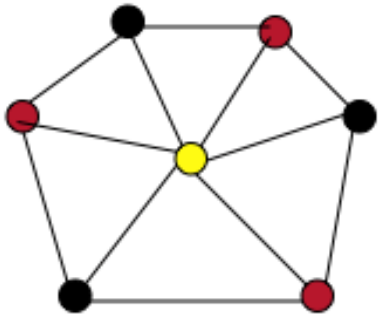
Grafo de Petersen



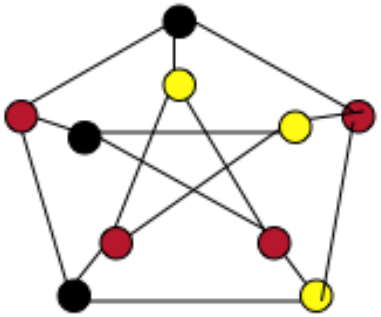
Resposta



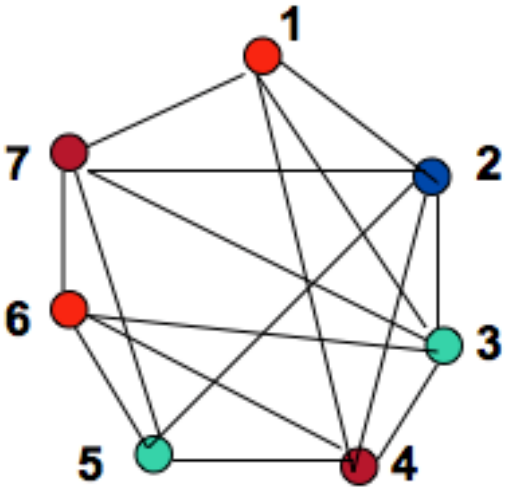
C_5



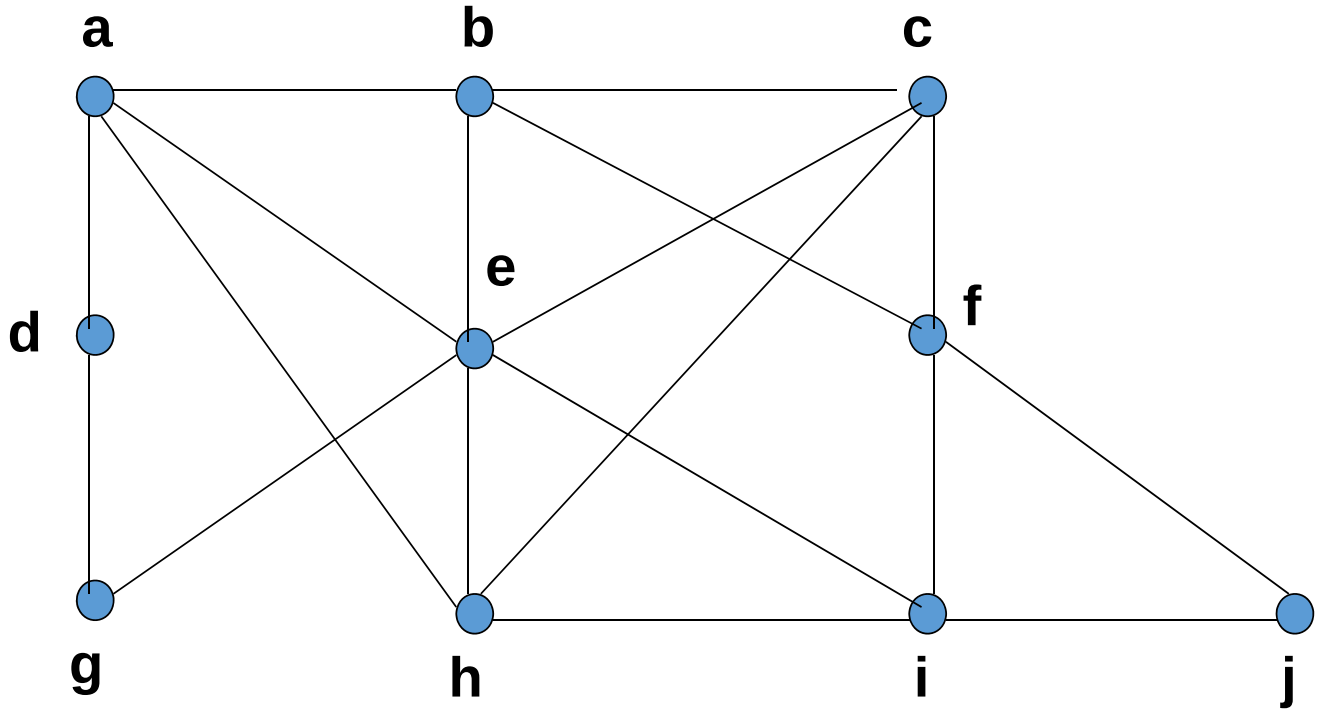
W_6



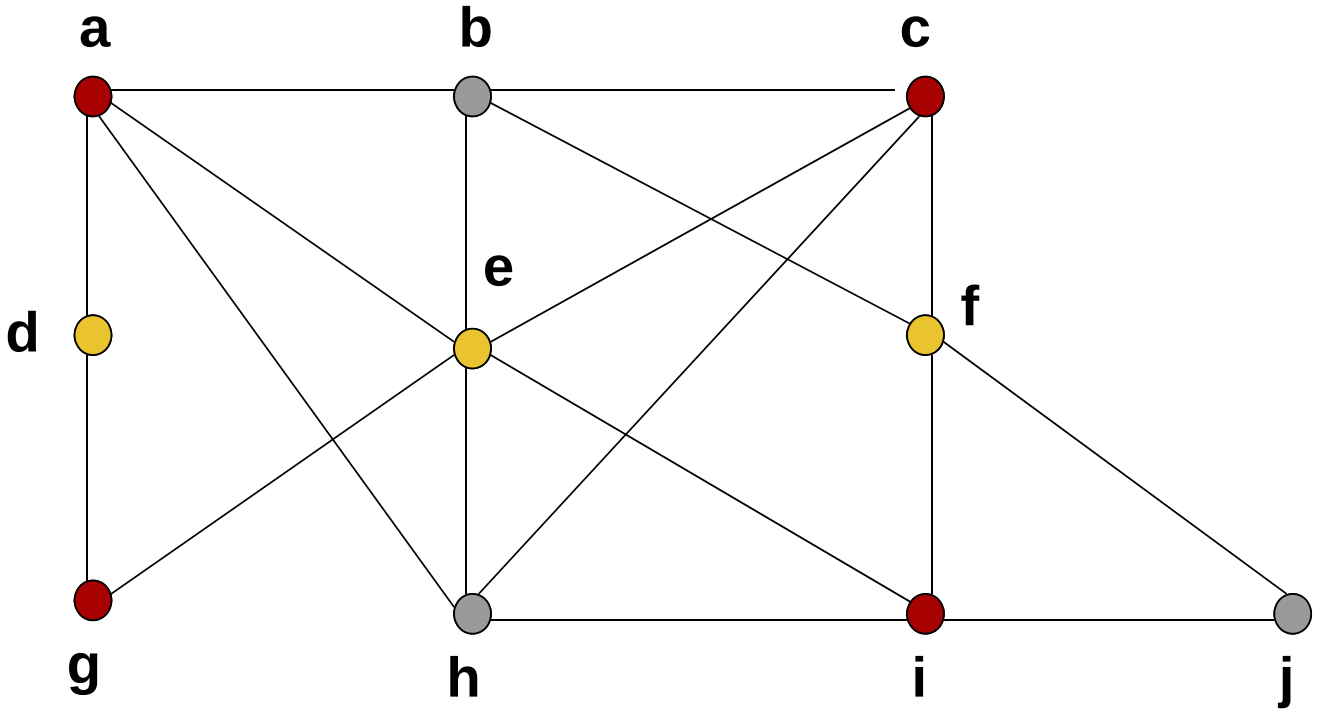
Grafo de Petersen



Exercício



Resposta



Ordem: e(6),a(4),b(4),c(4),f(4),h(4),i(4),d(2),g(2),j(2)

Referências

- http://www.lcad.icmc.usp.br/~nonato/ED/Coloracao/coloracao.html
- www.cin.ufpe.br/~if670/**Grafos**7-Coloracao.pptx