MC336 - PROJETO 5

Descobrir componentes conectados de um grafo

Escreva um predicado em Prolog que recebe uma lista de vertices, e uma lista de arestas, onde cada aresta é uma lista da forma `[a,b]` que indica que há uma aresta não direcionada entre vetices `a` e `b`. Note que se `[a,b]` aparece na lista, então `[b,a]` não aparece. O predicado retorna o número de componentes conectados do grafo, e em ordem crescente o número de vertices em cada componente.

O predicado é da forma `pred(V,A,N,L)` modo $++-$ onde V é a lista de vertices, A a lista de arestas, N o número de componentes conectados e L a lista, em ordem crescentes dos números de vertices em cada componente.

Uma vez que o seu predicado `pred` estiver correcto, inclua no seu arquivo o seguinte predicado:

```
main:-
        read(V),
        read(A),
        ignore(pred(V,A,N,L)),
        print(N),nl,
        print(L),nl.
```

que le os dados do arquivo de entrada (no stdin) e imprime o resultado no arquivo de saida (via stdout). O nome `main` é importante pois é o que o Susy chamará para executar seu programa.

Para calcular o número de componentes conexos, selecione um dos vertices não tratados e descubra usando busca em profundidade, todos os vertices cenectados a eles. Remova todos esses vertices, e recomeçe.