

MC336 - PROJETO 5

Imprimir a menor distancia de um vertice a todos os outros vertices de um grafo.

Escreva um predicado em Prolog que recebe uma lista de arestas `LA`, da forma

```
[[a, b, 5.3], [b, c, 12] ...]
```

e um vertice `V`.

A lista de arestas acima indica que há um arco entre os vertices a e b e o custo deste arco é 5.3, e que há um arco entre b e c com custo 12. Um arco aparece apenas uma vez na lista de entrada, e portanto não haverá um arco entre b e a na lista.

O predicado `pred(LA,V,Dist)`, modo `++-`, retorna em `Dist` uma lista das distancias entre o vertice `V` e os vértices do grafo, da forma

```
[ [a, 16.0], [b, 2.6], [c, Inf], [d, 0], ...]
```

ordenados por ordem crescente do nome do vertice, um por linha. A dupla `[c, Inf]` indica que não há um caminho entre o vertice inicial e o vertice c - e portanto o indica um custo ou distancia infinitos. A dupla `[d, 0]` indica que d era o vertice inicial.

Use o algoritmo de Dijkstra para calcular as distancias (projeto 2 do susy em Python).

Uma vez que o predicado `pred` esteja funcionando, inclua no seu arquivo o seguinte predicado:

```
main :- read(LA),
        read(V),
        ignore(pred(LA,V,Dist)),
        print(Dist),
        nl.
```

que le os dados do `stdin`, roda seu programa, e imprime a resposta no `stdout`. O nome `main` é importante pois esse é o predicado que o Susy vai chamar para rodar o seu programa.