

Descrição

Encontra o caminho mais curto entre dois vértices a e z de um grafo com pesos positivos.

Input

vértice: x

aresta: $e = \{i, j\}$

peso: $w(i, j) > 0$

rótulo de vértice: $L(x)$

Pseudocódigo

```
dijkstra(w,a,z,L){
    L(a) = 0
    for all vertices x != a
        L(x) = ∞
    T = set of all vertices

    while (z ∈ T){
        choose v ∈ T with min L(v)
        T = T \ {v}
        for each x ∈ T adjacent to v
            L(x) = min{L(x), L(v) + w(v,x)}
    }
}
```

Inicialização

- $L(a) = 0$ e para todos os outros vértices v , $L(v) = \infty$. Isso é feito no início porque, à medida que o algoritmo avança, a label da origem para cada nó v no gráfico será recalculada e finalizada quando a distância mais curta para z for encontrada
- T , uma lista com todos os nós do grafo. Ao fim da execução do algoritmo, T estará vazio.
- S (opcional), uma lista que indica quais nós o algoritmo já visitou. Ao fim da execução do algoritmo, S conterá todos os vértices do grafo.

Iterações

Enquanto T não estiver vazio, retirar o nó x , que ainda não está em S , de T com o menor $L(x)$. Na primeira execução, o nó de origem a será escolhido porque $L(a)$ foi inicializado com 0. Na próxima execução, o próximo nó com o menor valor de w é escolhido.

Adicione o nó v ao set S , para indicar que x foi visitado.

Atualize os valores L dos nós adjacentes do nó atual x da seguinte forma: para cada novo nó adjacente v , se $L(x) + peso(x, v) < L(v)$, há uma nova distância mínima encontrada para v , então

atualize $L(x)$ para o novo valor de distância mínima.
Caso contrário, nenhuma atualização será feita para $L(x)$.