Algoritmo de Dijkstra para obter um Caminho mais Curto

Teoria dos Grafos – 2025-2

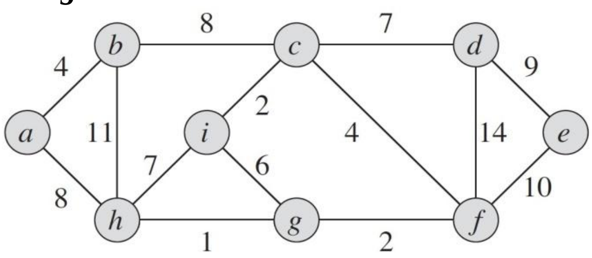
Prof. Roberto C. de Araujo

1. Caminho mais curto em um grafo

Um caminho mais curto de um vértice ***x*** até um vértice ***y*** em um grafo G foi definido como a quantidade de arestas que ocorrem em um caminho de ***x*** até ***y*** que tenha a menor quantidade de arestas possível. No entanto, o conceito de caminho mais curto pode ser adaptado a situações em que as arestas do grafo possuem um "***custo***" associado a elas. O custo de uma aresta ***a*** é um valor não negativo que pode ser entendido como uma quantia a ser paga ao se usar a aresta ***a*** uma vez em um passeio.[1]

Um ***caminho mais curto*** em um grafo ponderado é definido como o caminho cuja soma dos custos das arestas que ocorrem no caminho é o menor possível.

Exercício: No grafo abaixo, qual é o caminho mais curto de ***g*** até ***d***?



2. Caminho mais curto em grafo orientado

Um ***grafo orientado*** (também chamado de ***dígrafo***) é um grafo no qual as "arestas" têm um sentido (nesse caso, usamos o nome ***arco*** no lugar de aresta para indicar a existência do sentido). Assim, um arco ***xy*** com ***cauda x*** e ***cabeça y*** pode ser usado para acessar ***y*** a partir de ***x***, mas não pode ser usado para acessar ***x*** a partir de ***y***. O ***grau de entrada*** de um vértice v, $g^-(v)$, é a quantidade de arcos com v como cabeça; o ***grau de saída*** de um vértice v, $g^+(v)$, é a quantidade de arcos com v como cauda.

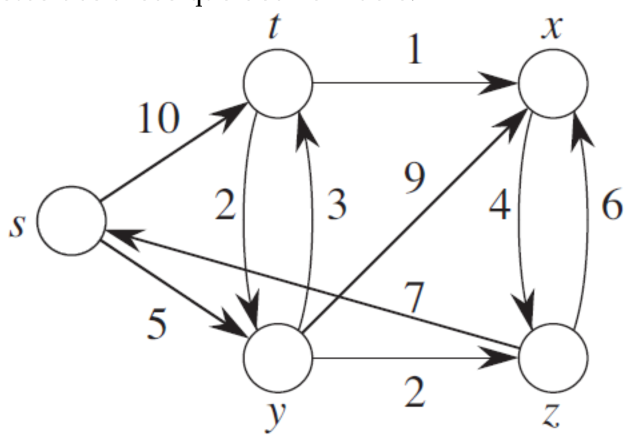
Nos grafos orientados, um passeio de um vértice ***a*** até um vértice ***b*** deve ter todos os seus arcos orientados conforme a sequência de vértices do passeio.

Um ***caminho mais curto*** em um dígrafo ponderado é definido como um caminho cuja soma dos custos dos arcos que ocorrem no caminho é o menor possível.

[1] Alguns autores denominam de ***grafo ponderado*** um grafo com custos (ou pesos) associados às suas arestas.

3. O algoritmo de Dijkstra

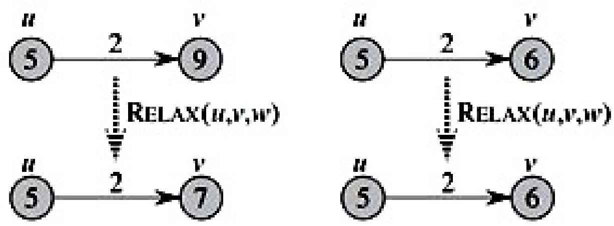
Dado um grafo orientado com custos nos seus arcos, obter um "caminho mais curto" de um vértice inicial s até todos os demais vértices do grafo.
O custo de um caminho é definido como a soma dos custos dos arcos que ocorrem dele.




```
DIJKSTRA(G, w, s)
  INITIALIZE-SINGLE-SOURCE(G, s)
  S = ∅
  Q = G.V
  while Q ≠ ∅
      u = EXTRACT-MIN(Q)
      S = S ∪ {u}
      for each vertex v ∈ G.Adj[u]
          RELAX(u, v, w)
```

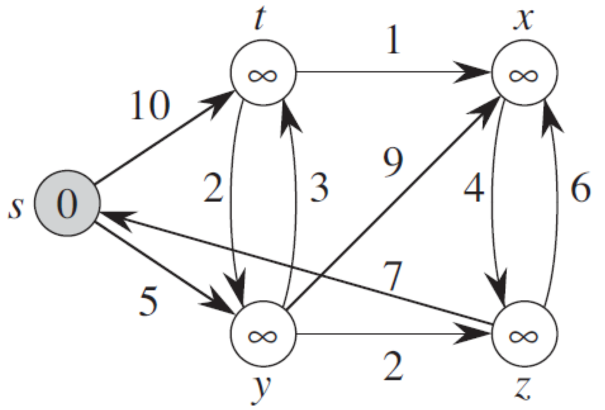
```
INITIALIZE-SINGLE-SOURCE(G, s)
  for each vertex v ∈ G.V
      v.d = ∞
      v.π = NIL
  s.d = 0
```

```
RELAX(u, v, w)
  if v.d > u.d + w(u, v)
      v.d = u.d + w(u, v)
      v.π = u
```

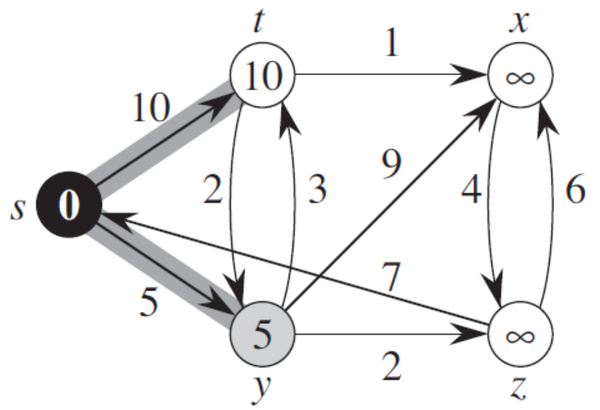


Simulação do algoritmo de Dijkstra

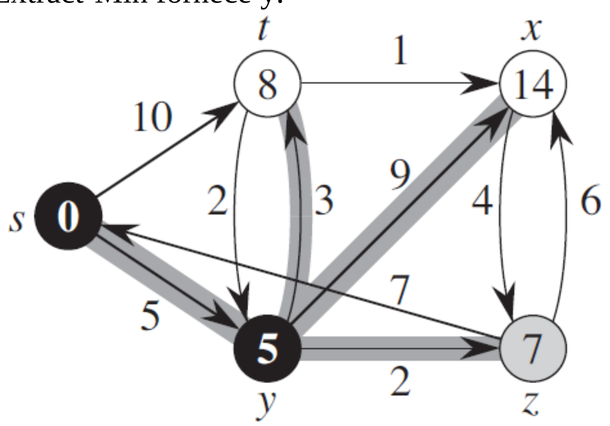
a) Inicialização:



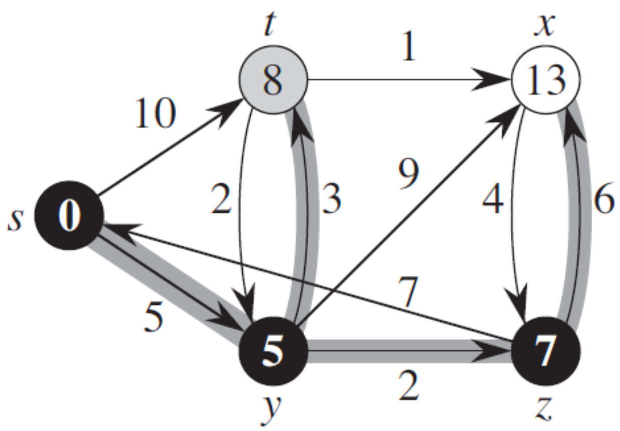
b) Extract-Min fornece s:



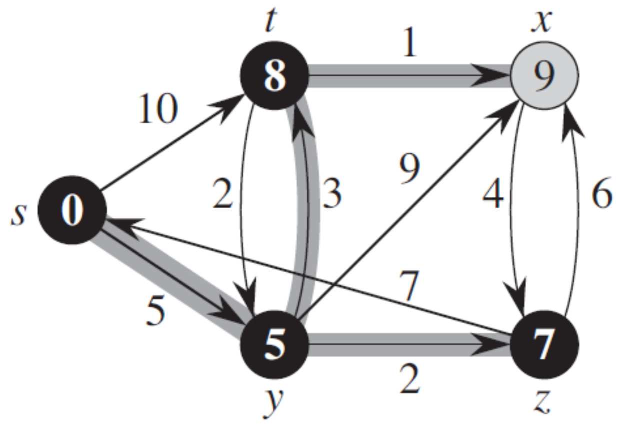
c) Extract-Min fornece y:



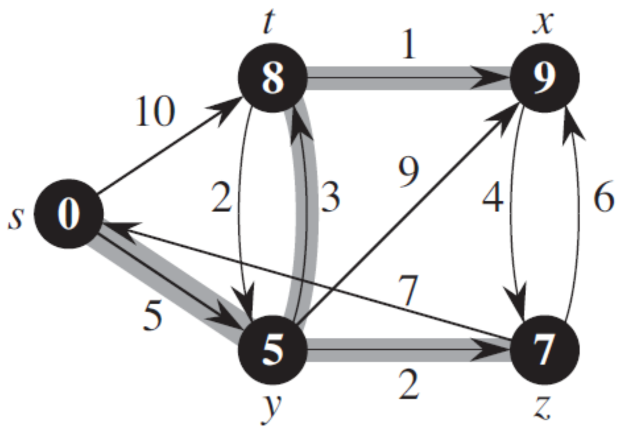
d) Extract-Min fornece z:



e) Extract-Min fornece t:



f) Extract-Min fornece x:



4. Exercício

Considerando o grafo orientado abaixo com custos associados aos seus arcos), simule, passo a passo, conforme feito em aula, a execução do algoritmo de Dijkstra para obtenção dos caminhos de custo mínimo a partir do vértice ***s***.