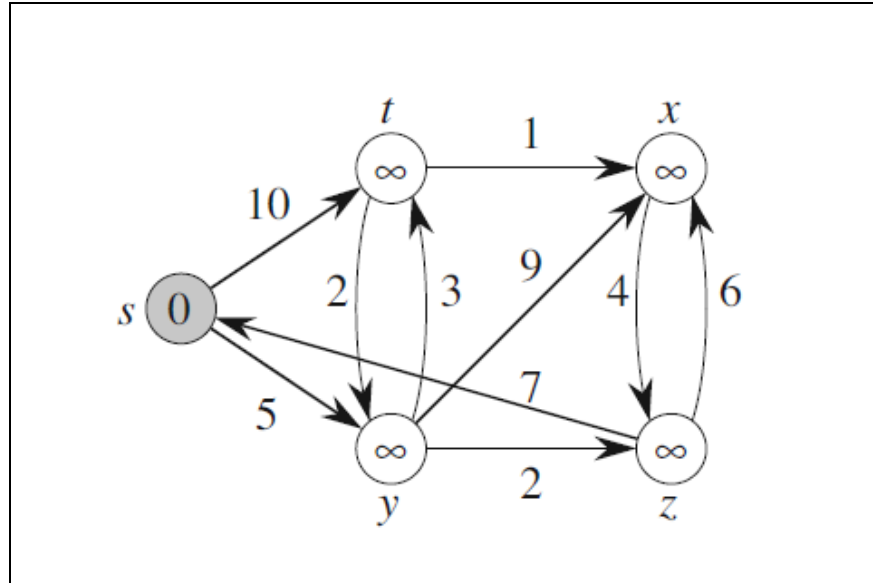


## Zadaci

- Implementirati graf prikazan na slici 1. Modifikovati klasu Vertex da podrži prikaz težinskih usmerenih grafova. Pored klase Vertex napraviti i klasu Edge koja predstavlja ivicu grafa. Ova klasa minimalno treba da sadrži polja *source*, *destination* i *weight*.

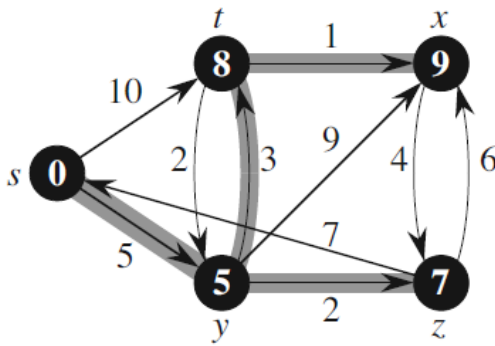


Slika1 – Ilustracija usmerenog težinskog grafa.

- Implementirati Dijkstra algoritam. Pseudo kodovi potrebnih funkcija su dati na slici 2., kao i izgled rezultujućeg grafa.

<pre> INITIALIZE-SINGLE-SOURCE(<math>G, s</math>) 1  <b>for</b> each vertex <math>v \in G.V</math> 2      <math>v.d = \infty</math> 3      <math>v.\pi = \text{NIL}</math> 4  <math>s.d = 0</math> </pre>	<pre> DIJKSTRA(<math>G, w, s</math>) 1  INITIALIZE-SINGLE-SOURCE(<math>G, s</math>) 2  <math>S = \emptyset</math> 3  <math>Q = G.V</math> 4  <b>while</b> <math>Q \neq \emptyset</math> 5      <math>u = \text{EXTRACT-MIN}(Q)</math> 6      <math>S = S \cup \{u\}</math> 7      <b>for</b> each vertex <math>v \in G.Adj[u]</math> 8          RELAX(<math>u, v, w</math>) </pre>
<pre> RELAX(<math>u, v, w</math>) 1  <b>if</b> <math>v.d &gt; u.d + w(u, v)</math> 2      <math>v.d = u.d + w(u, v)</math> 3      <math>v.\pi = u</math> </pre>	

Rezultat algoritma:



Slika2 – Pseudokod funkcija potrebnih za Dijkstra algoritam.

3. Napraviti funkciju za nasumično generisanje usmerenog težinskog grafa i iskoristiti je za testiranje implementiranog algoritma.