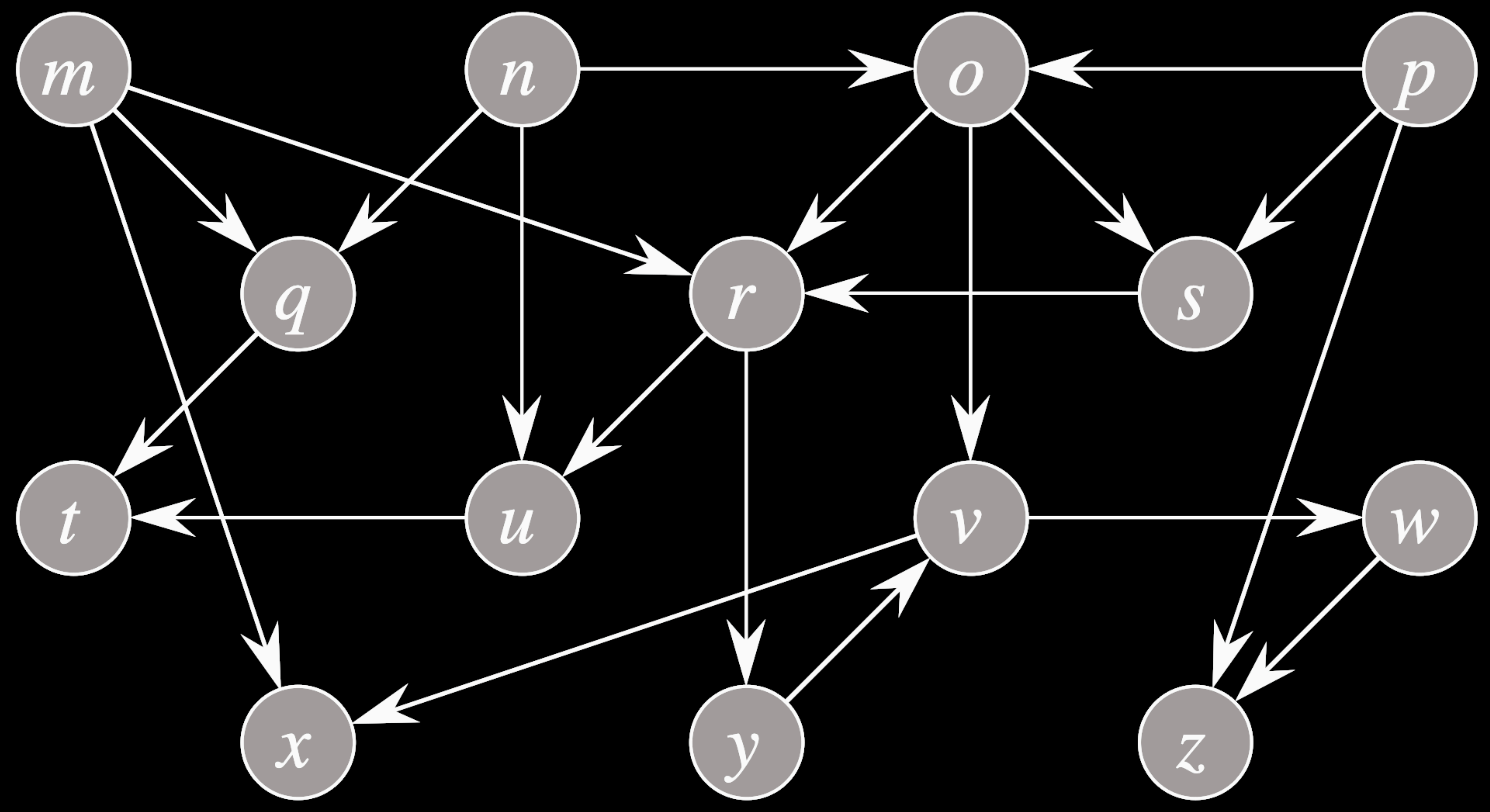
Algorithm Design and Analysis

David N. JANSEN, Bohua ZHAN
名 姓

算法设计与分析

詹博华，杨大卫

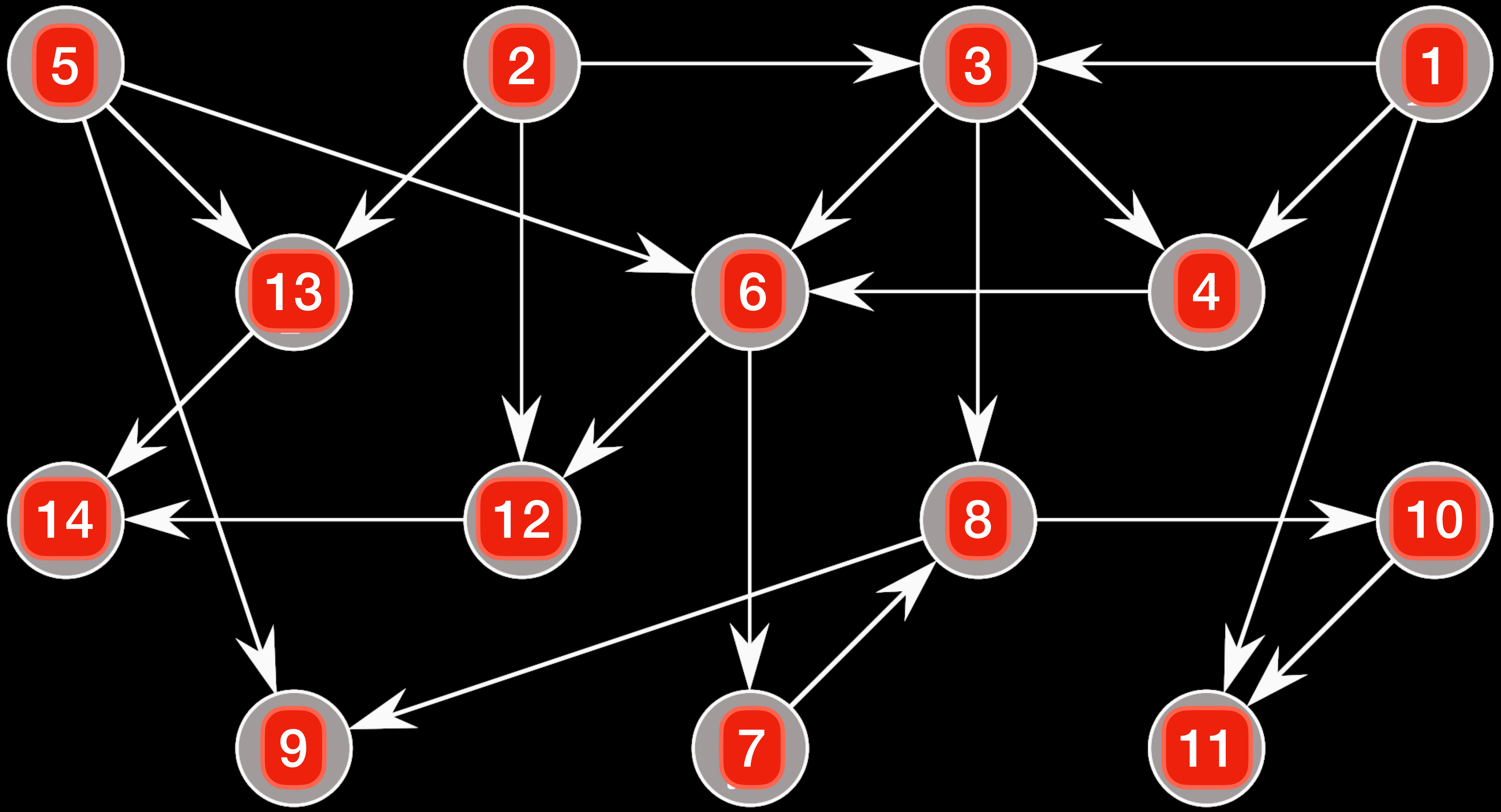
Exercise 22.4-1



Show the ordering of vertices produced by TOPOLOGICAL-SORT when it is run on the dag right. Assume that the for loop for lines 5–7 of DFS considers the vertices in alphabetical order, and assume that each adjacency list is ordered alphabetically.

给出算法 TOPOLOGICAL-SORT 运行于图 22-8 上时所生成的结点次序。假定深度优先搜索算法的第5～7行的 **for** 循环是以字母表顺序依次处理每个结点，并假定每条邻接链表皆以字母表顺序对里面的结点进行了排序。


m
n
p
5
2
1
q
r
o
13
6
3
t
u
y
s
14
12
7
4
v
8
w
x
10
9
z
11


5
2
3
1
13
6
4
14
12
8
10
9
7
11

Exercise 23.1-7

Argue that if all edge weights of a graph are positive, then any subset of edges that connects all vertices and has minimum total weight must be a tree. Give an example to show that the same conclusion does not follow if we allow some weights to be nonpositive.

证明：如果一个图的所有边的权重都是正值，则任意一个连接所有结点且总权重最小的一个边集合必然形成一棵树。另外，请举出例子来证明：如果允许某些边的权重为负值，则该论断不成立。

- A tree is a connected, acyclic graph.
- Assume given a weighted graph ($G = (V, E)$, $w: E \to \mathbb{R}^+$) and a subset $A \subseteq E$ such that (V, A) is connected and the sum of the weights in A is minimal among all such subsets.
- By assumption (V, A) is connected.
It remains to be proven that (V, A) is acyclic.

- 一棵树是一个联通的无环的图。
- 设定一个权重的图 ($G = (V, E)$, $w: E \to \mathbb{R}^+$) 和子集 $A \subseteq E$ 这样 (V, A) 是联通的，并且在 A 边的权重的总体是最小的
在所有这样的子集中。
- 通过假设 (V, A) 是连通的。
还待证明 (V, A) 是无环的。

- It remains to be proven that (V, A) is acyclic.

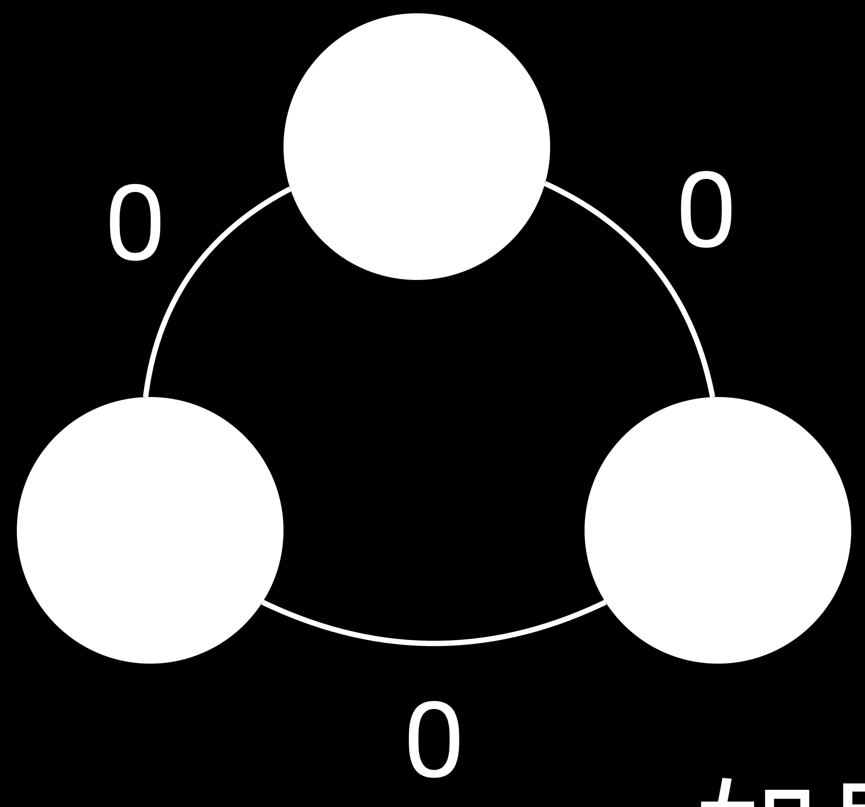
- But if (V, A) would contain a cycle, one could remove any edge (u, v) in the cycle and still have a connected graph (V, $A \setminus \{(u, v)\}$). As $w((u, v)) > 0$, we must have that $A \setminus \{(u, v)\}$ is a cheaper subset of E that keeps the graph connected. Contradiction!

- 还待证明 (V, A) 是无环的。

- 但如果 (V, A) 包含一个环路，则可以移除环路中的任何边 (u, v)，并且仍然有一个连通图 (V, $A \setminus \{(u, v)\}$)。当 $w((u, v)) > 0$ 时，必须 $A \setminus \{(u, v)\}$ 是 E 的一个更便宜的子集，它使图保持连通。矛盾！

- If there are zero-weight edges, they can be added without changing the cost of the subset *A*.

- 如果有权重为0的边，增加这条边跟边的子集的权重总体没有影响。

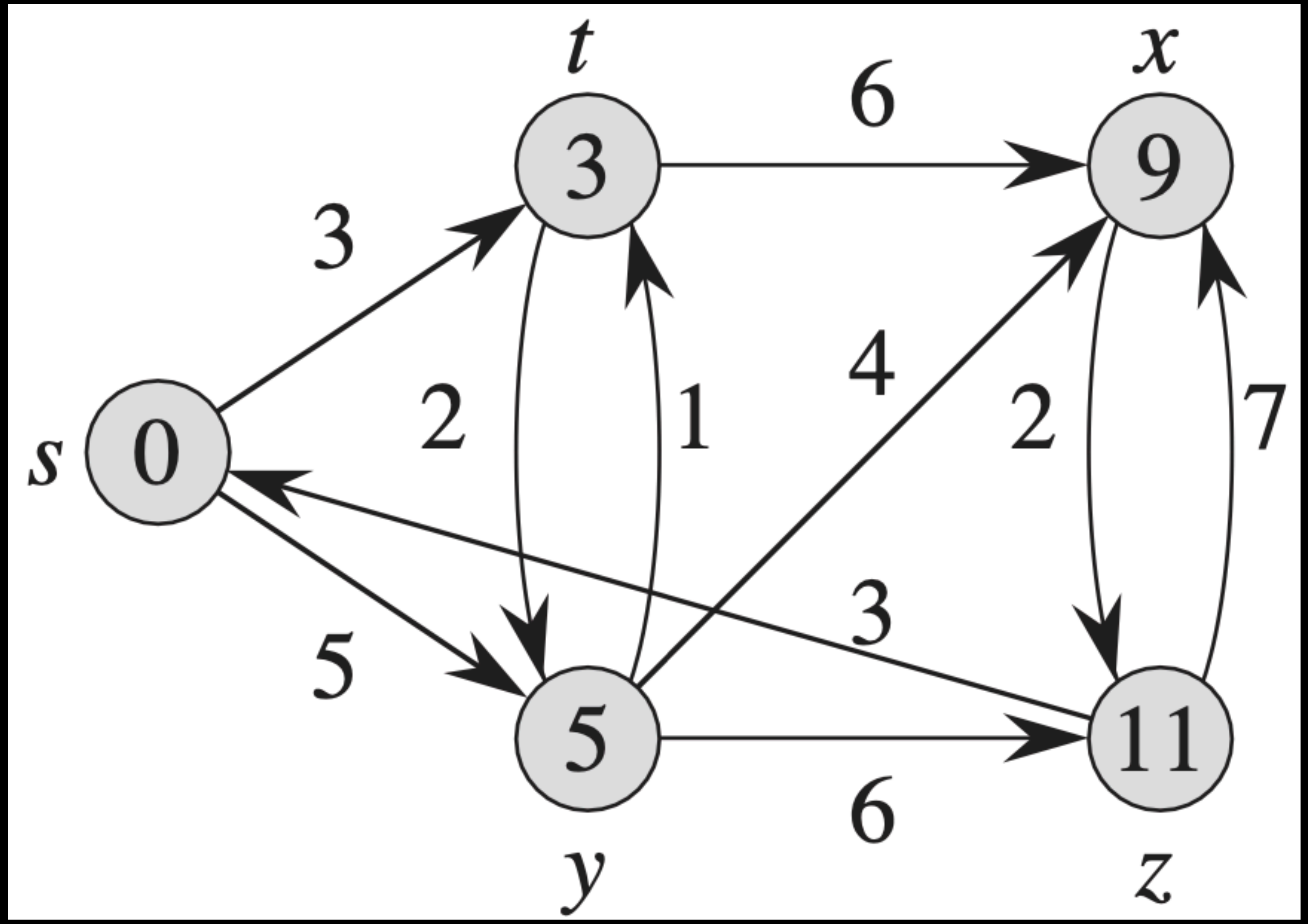


- If there are negative-weight edges, every negative-weight edge is in a minimal-weight set.

- 如果有负值的权重的边，它们必须在所有的最小权重的边集。

Exercise 24.3-1

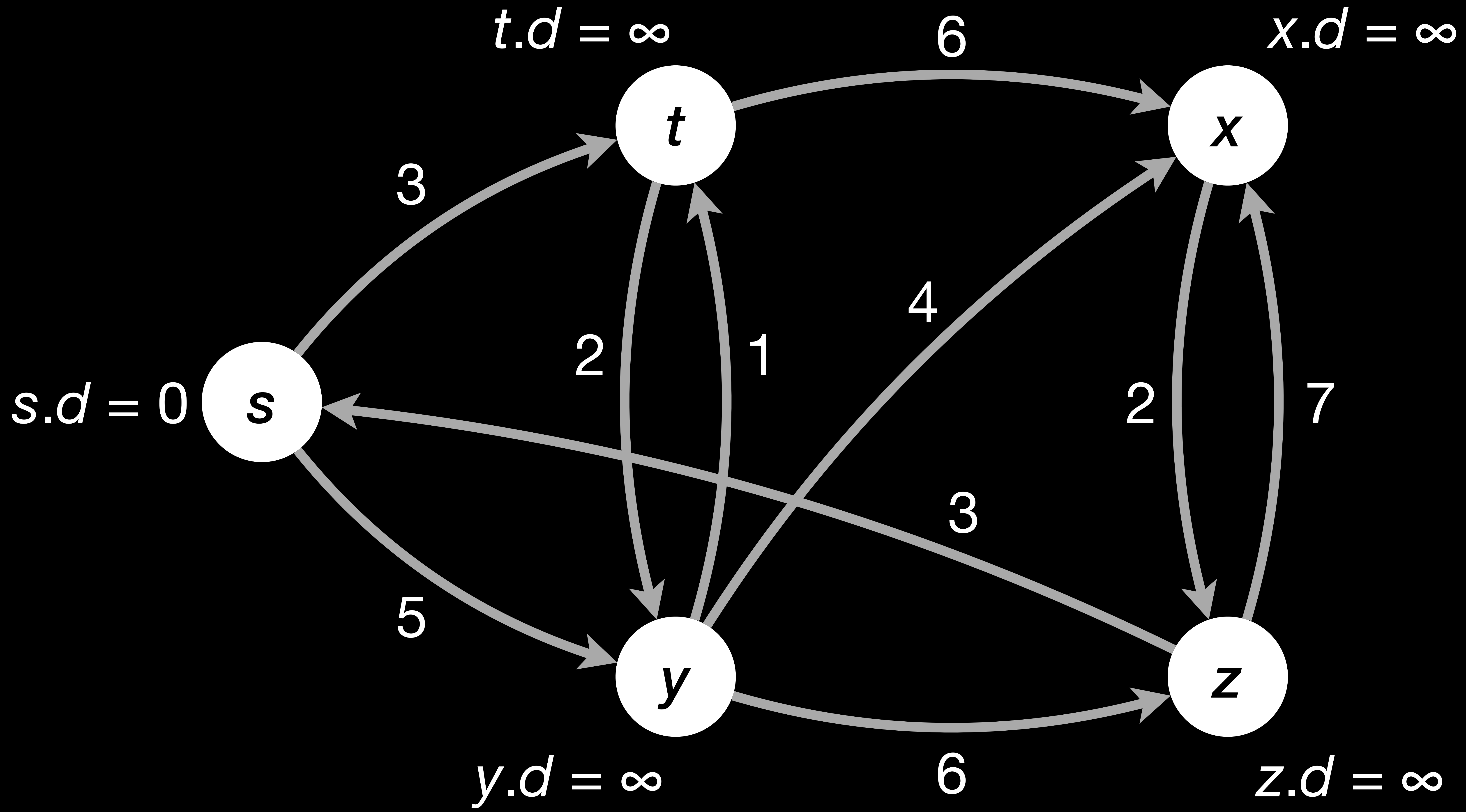
Run Dijkstra's algorithm on the directed graph to the right, first using vertex *s* as the source and then using vertex *z*.
In the style of Fig. 24.6, show the *d* and π values and the vertices in set *S* after each iteration of the **while** loop.



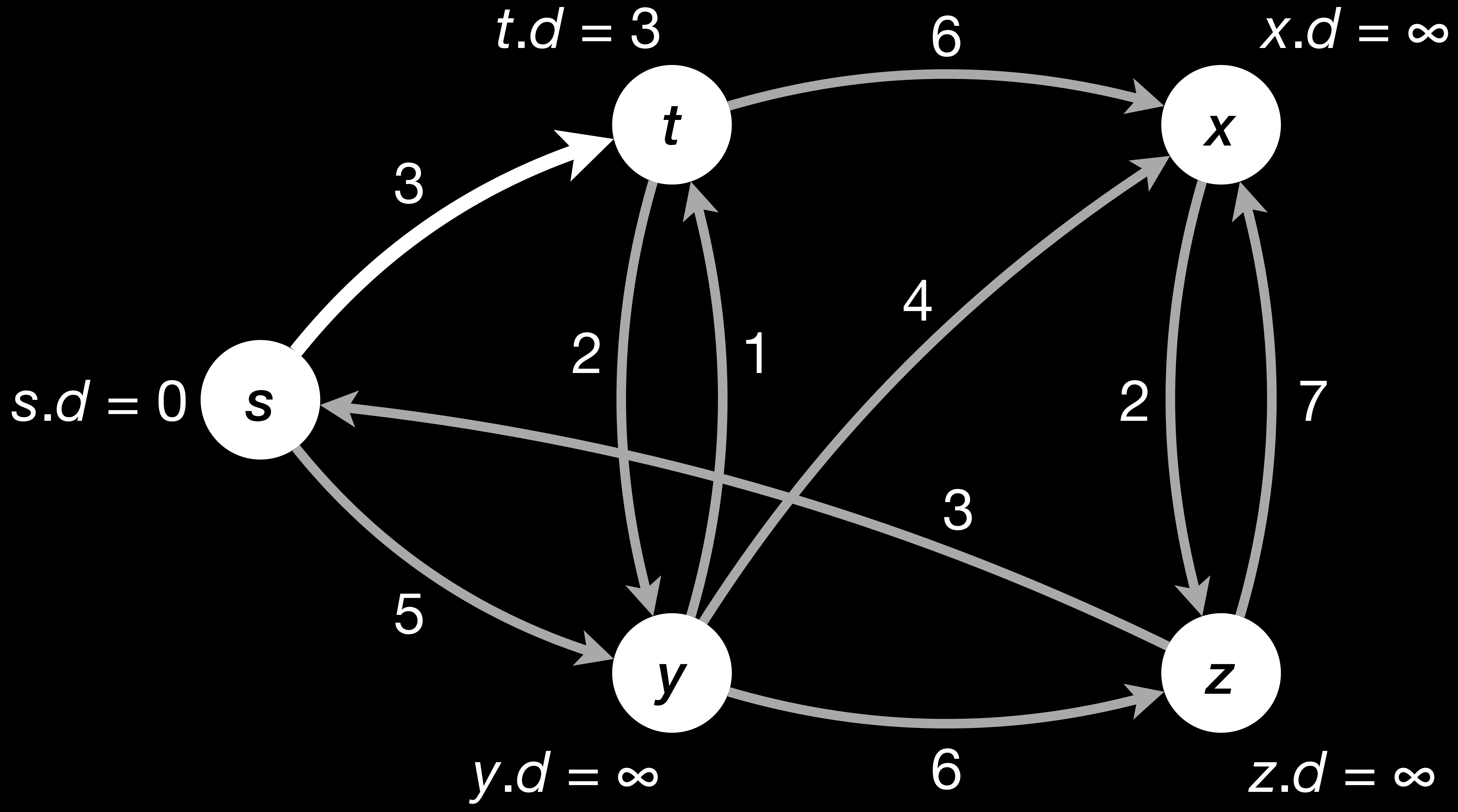
在图 24-2 上运行 Dijkstra 算法，第一次使用结点 *s* 作为源结点，第二次使用结点 *z* 作为源结点。以类似于图 24-6 的风格，给出每次 **while** 循环后的 *d* 值和 π 值，以及集合 *S* 中的所有结点。

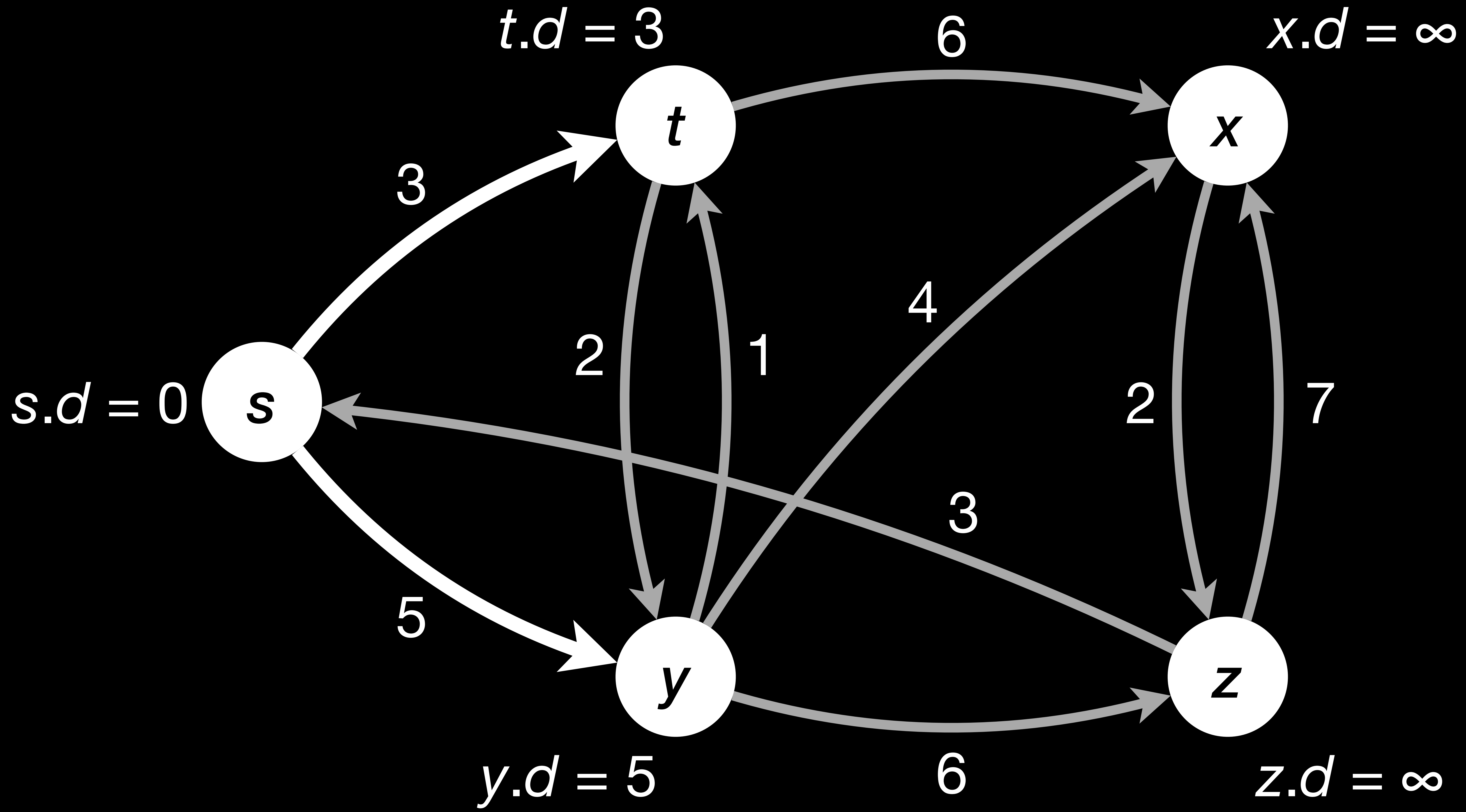
Bellman–Ford

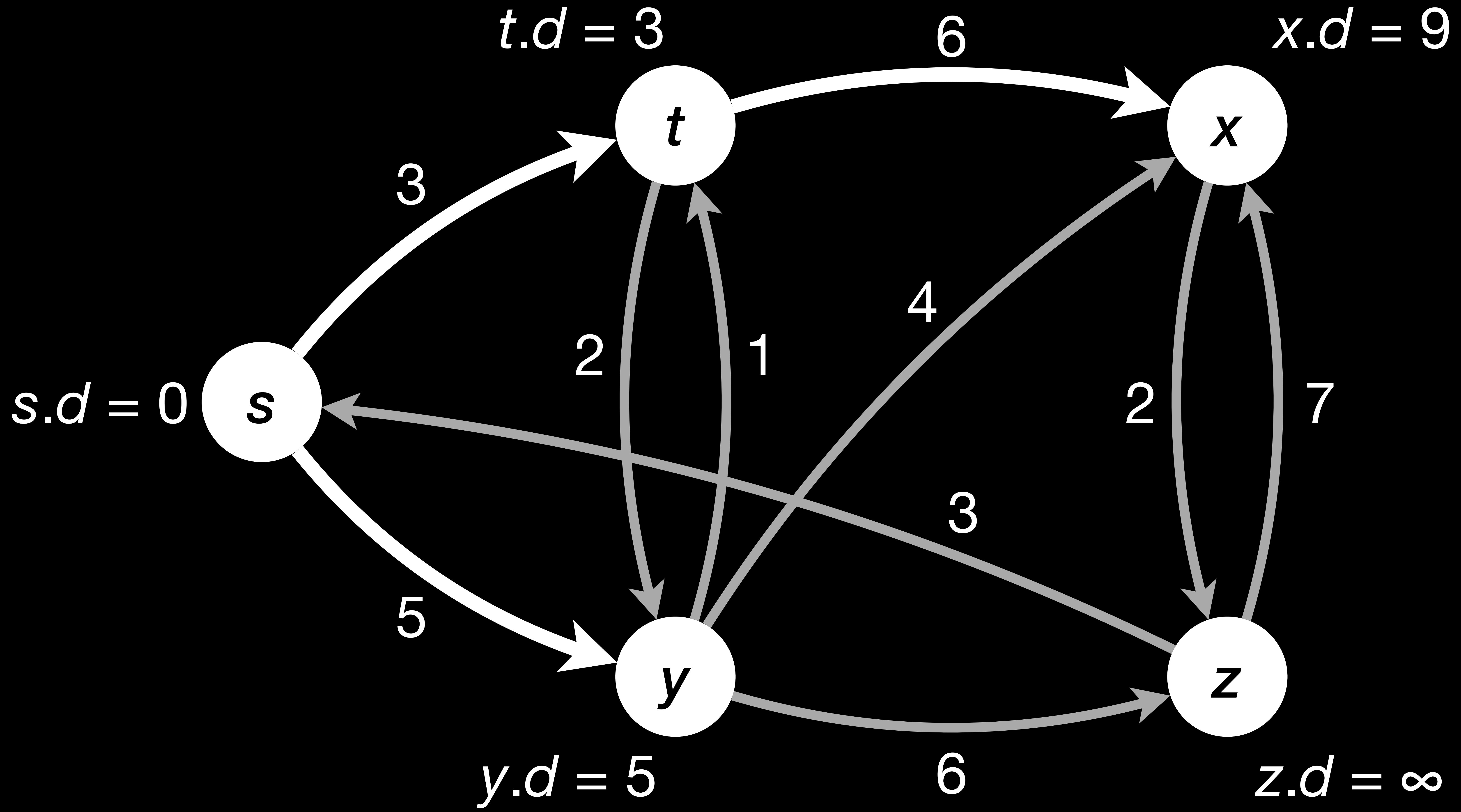
source = s



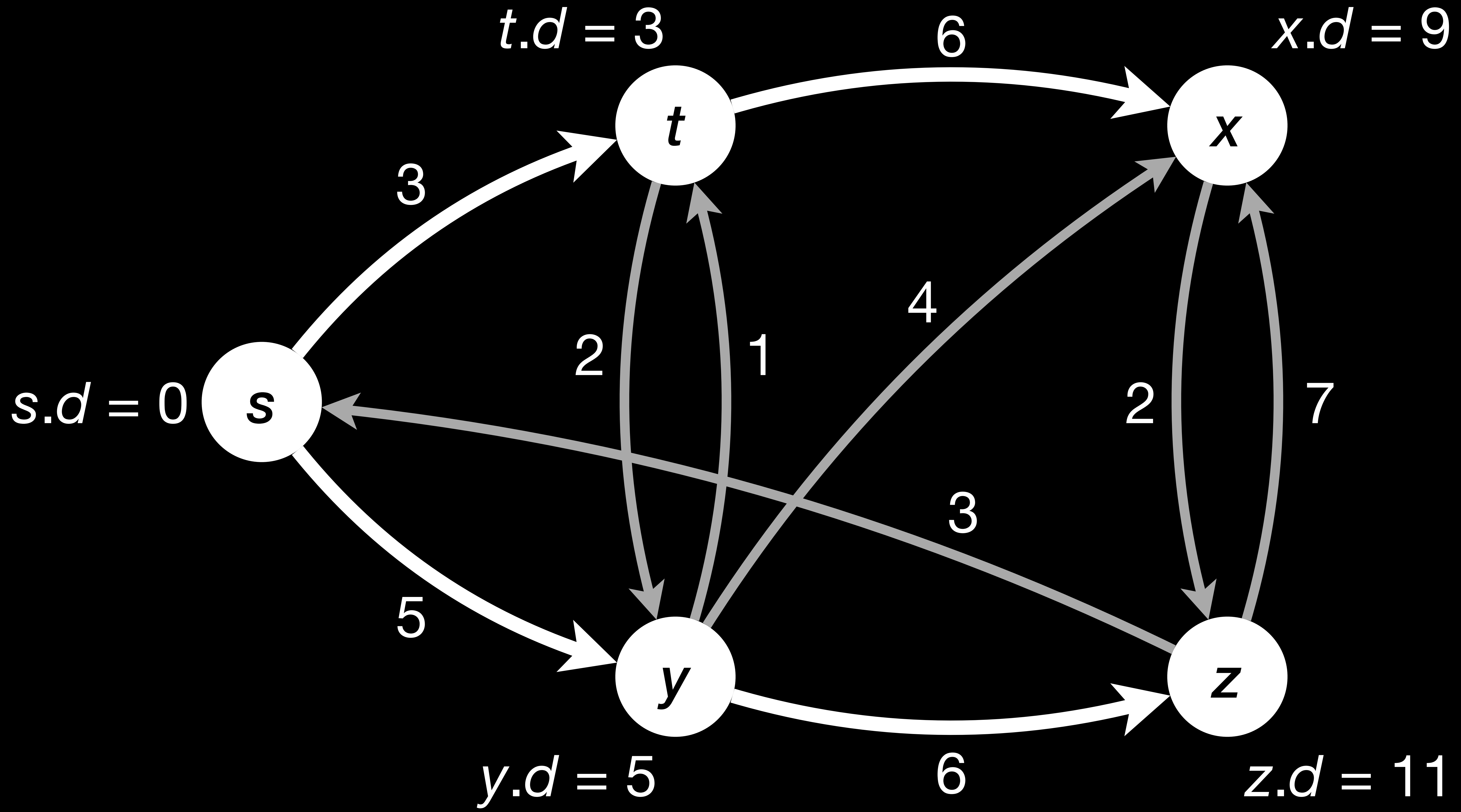
$t.d = \infty$
6
$x.d = \infty$
t
x
3
4
2
1
2
7
$s.d = 0$
s
3
5
y
z
$y.d = \infty$
6
$z.d = \infty$


t.d = 3
6
x.d = ∞
t
x
3
2
1
4
s.d = 0
s
2
7
3
5
y
z
y.d = ∞
6
z.d = ∞


t.d = 3
6
x.d = ∞
t
x
3
2
1
4
s.d = 0
s
2
7
3
5
y
z
y.d = 5
6
z.d = ∞



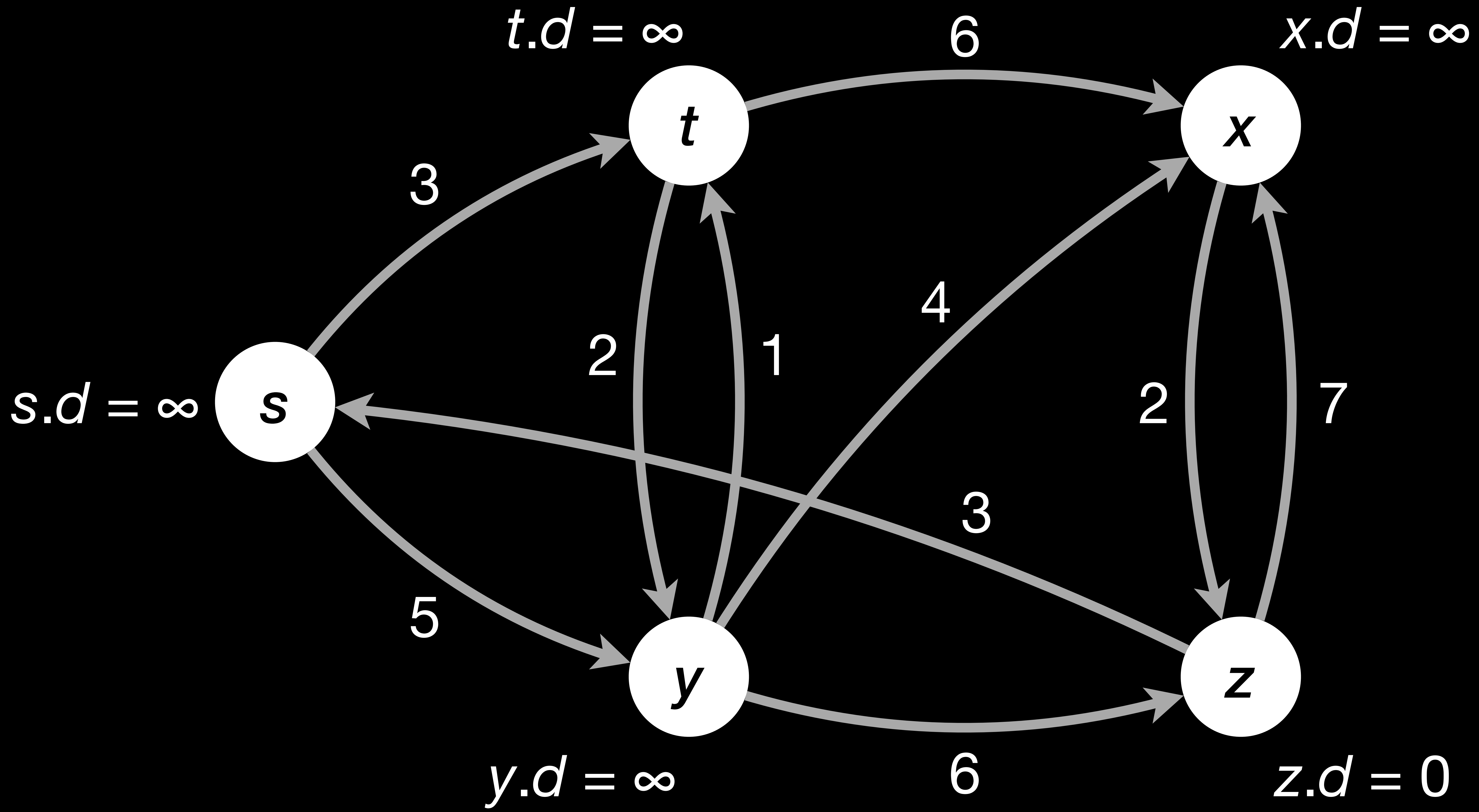
t.d = 3
6
x.d = 9
t
x
3
2
1
4
s.d = 0
s
2
7
3
5
y
z
y.d = 5
6
z.d = ∞

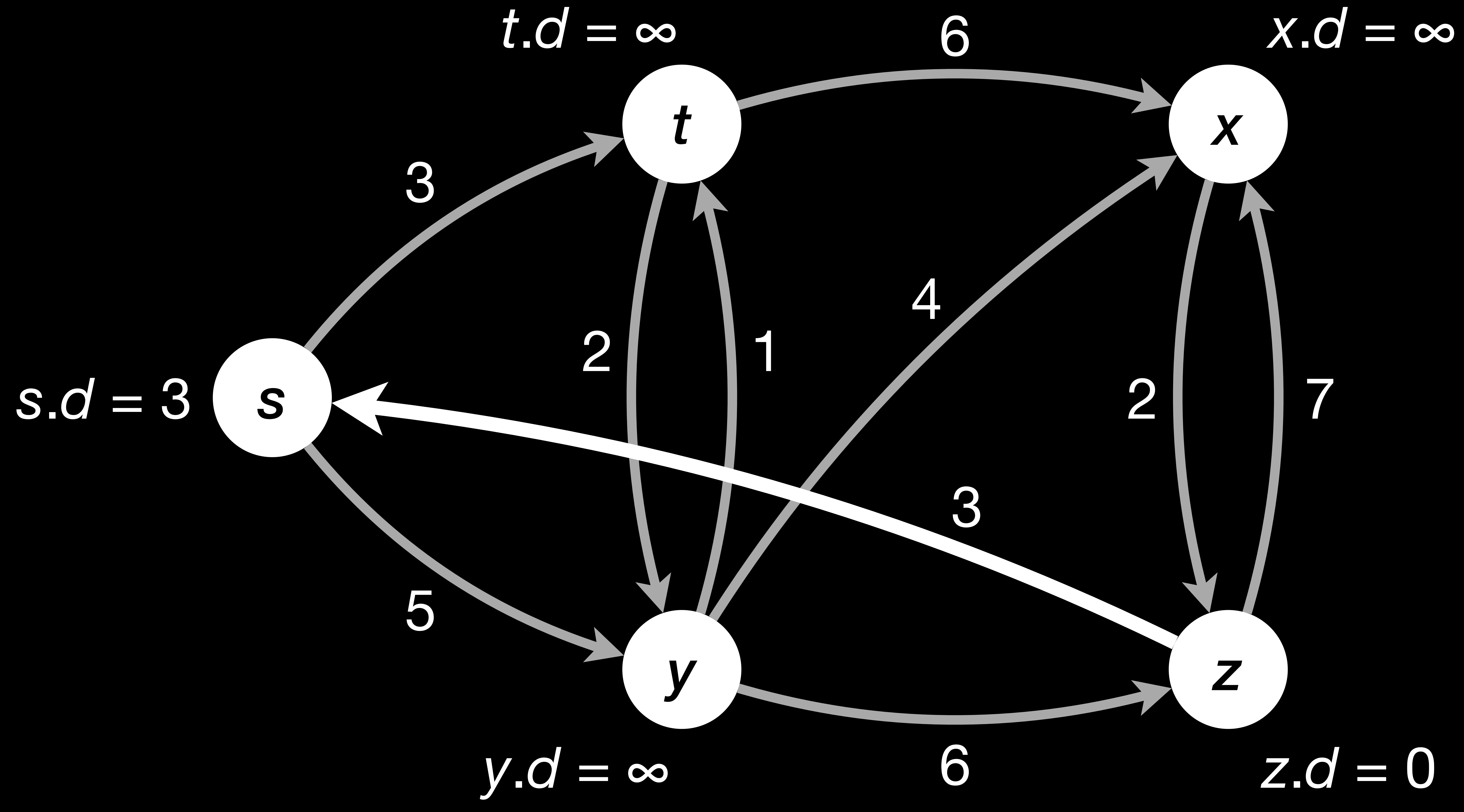


t.d = 3
6
x.d = 9
t
x
3
2
1
4
s.d = 0
s
2
7
3
5
y
z
y.d = 5
6
z.d = 11

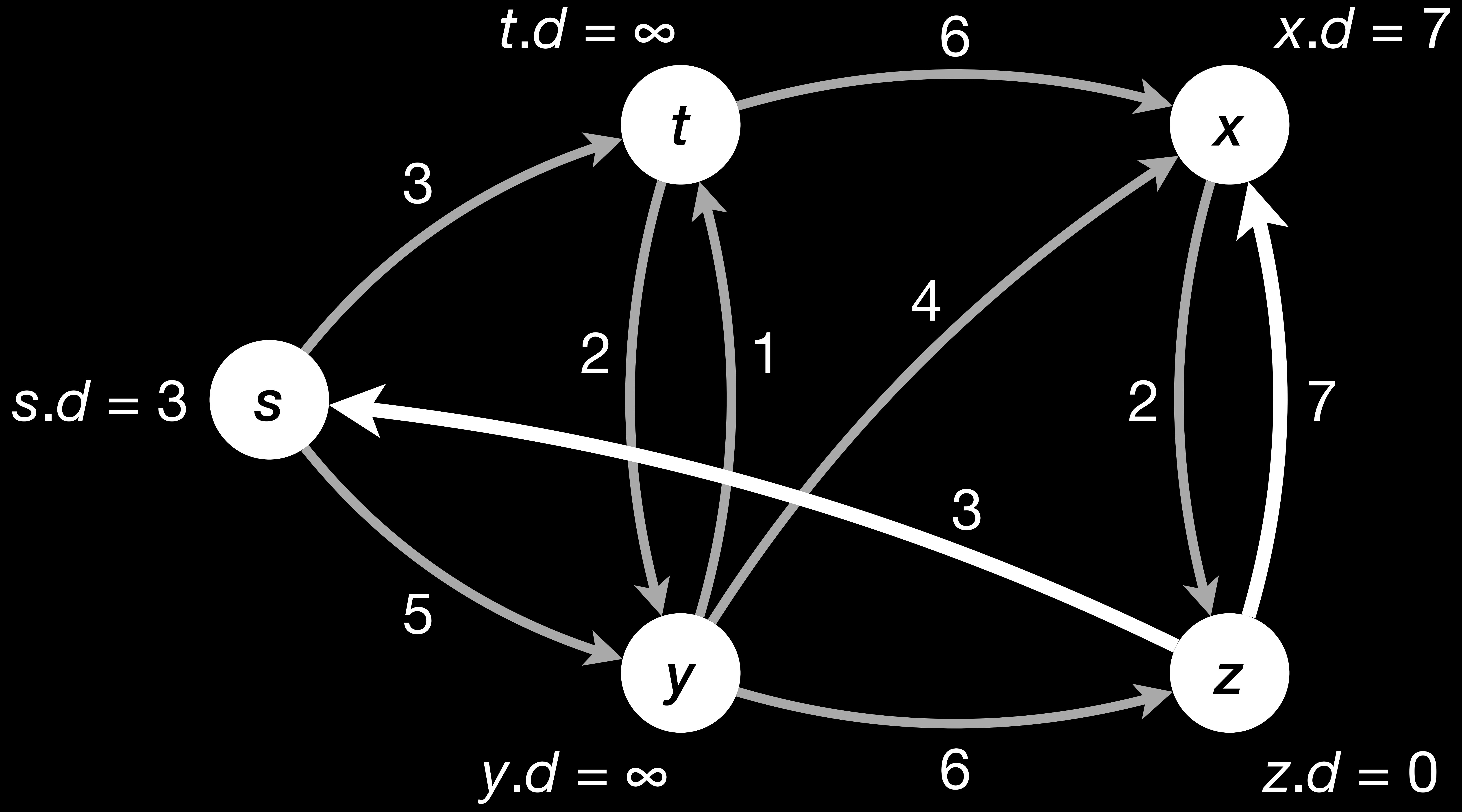
Bellman–Ford

source = z

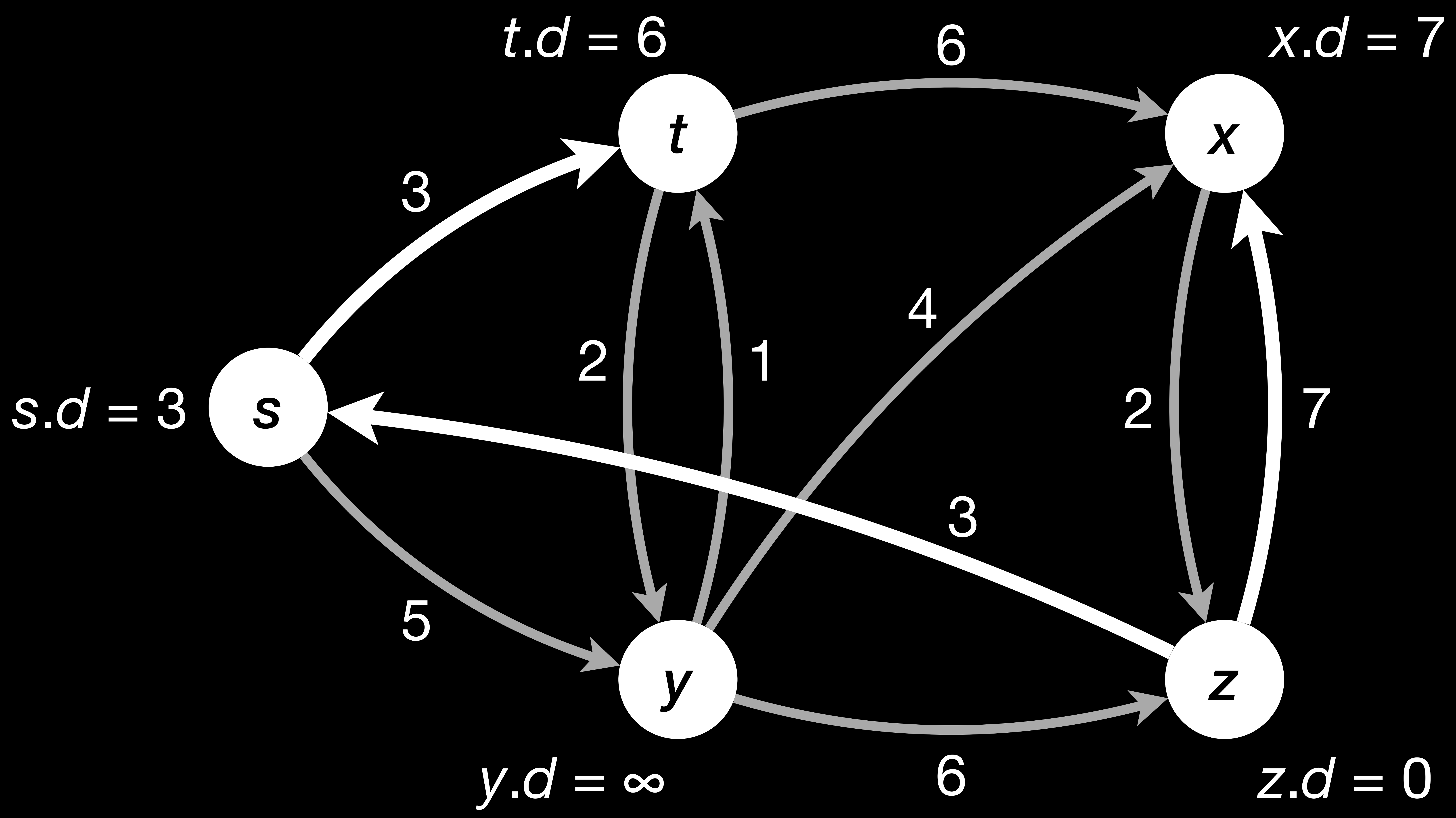

t.d = ∞
6
x.d = ∞
t
x
3
4
2
1
s.d = ∞
s
2
7
3
5
y
z
y.d = ∞
6
z.d = 0



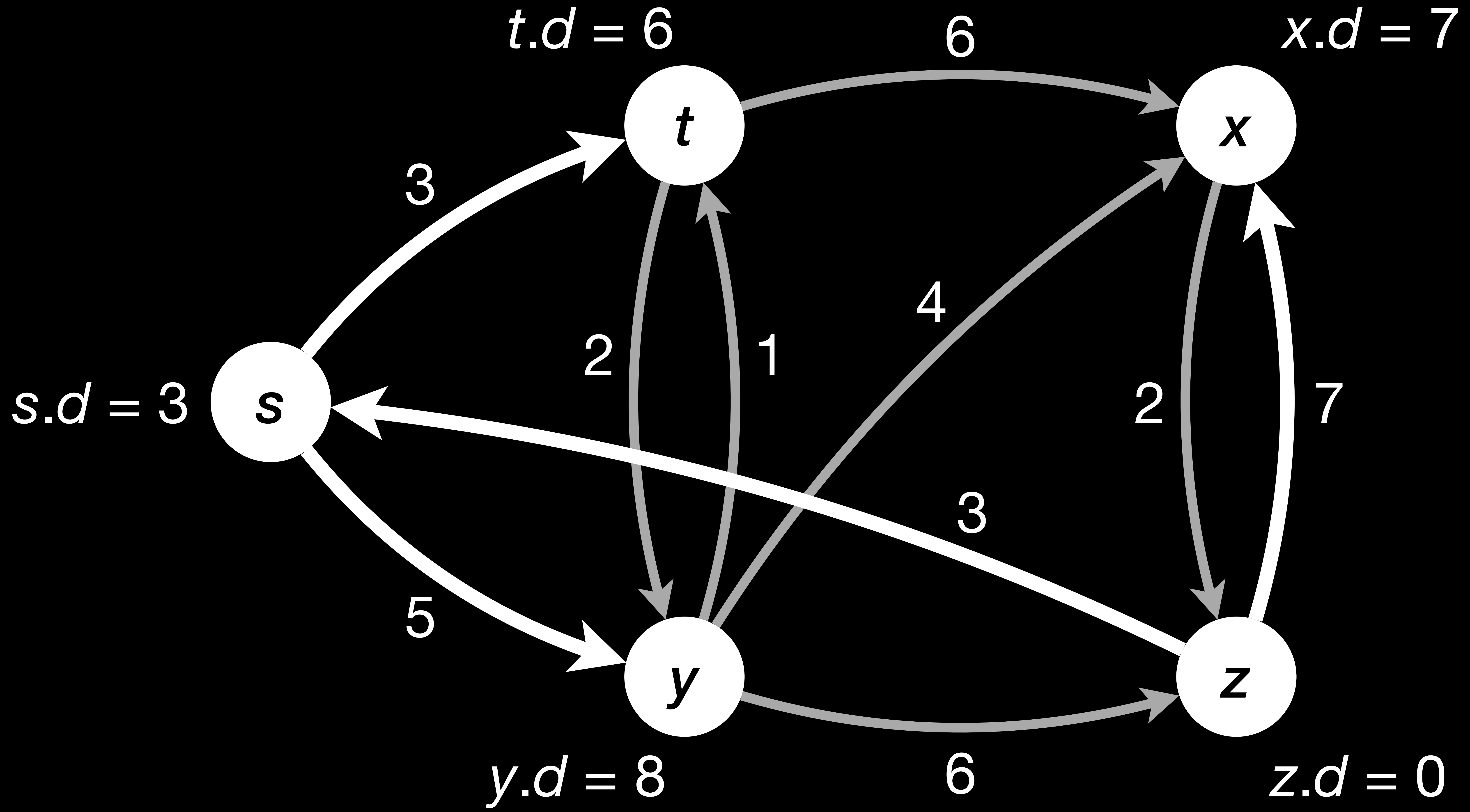
$t.d = \infty$
6
$x.d = \infty$
t
x
3
2
1
4
$s.d = 3$
s
2
7
3
5
y
z
$y.d = \infty$
6
$z.d = 0$



$t.d = \infty$
6
$x.d = 7$
t
x
3
4
2
1
2
7
$s.d = 3$
s
3
5
y
z
6
$y.d = \infty$
$z.d = 0$



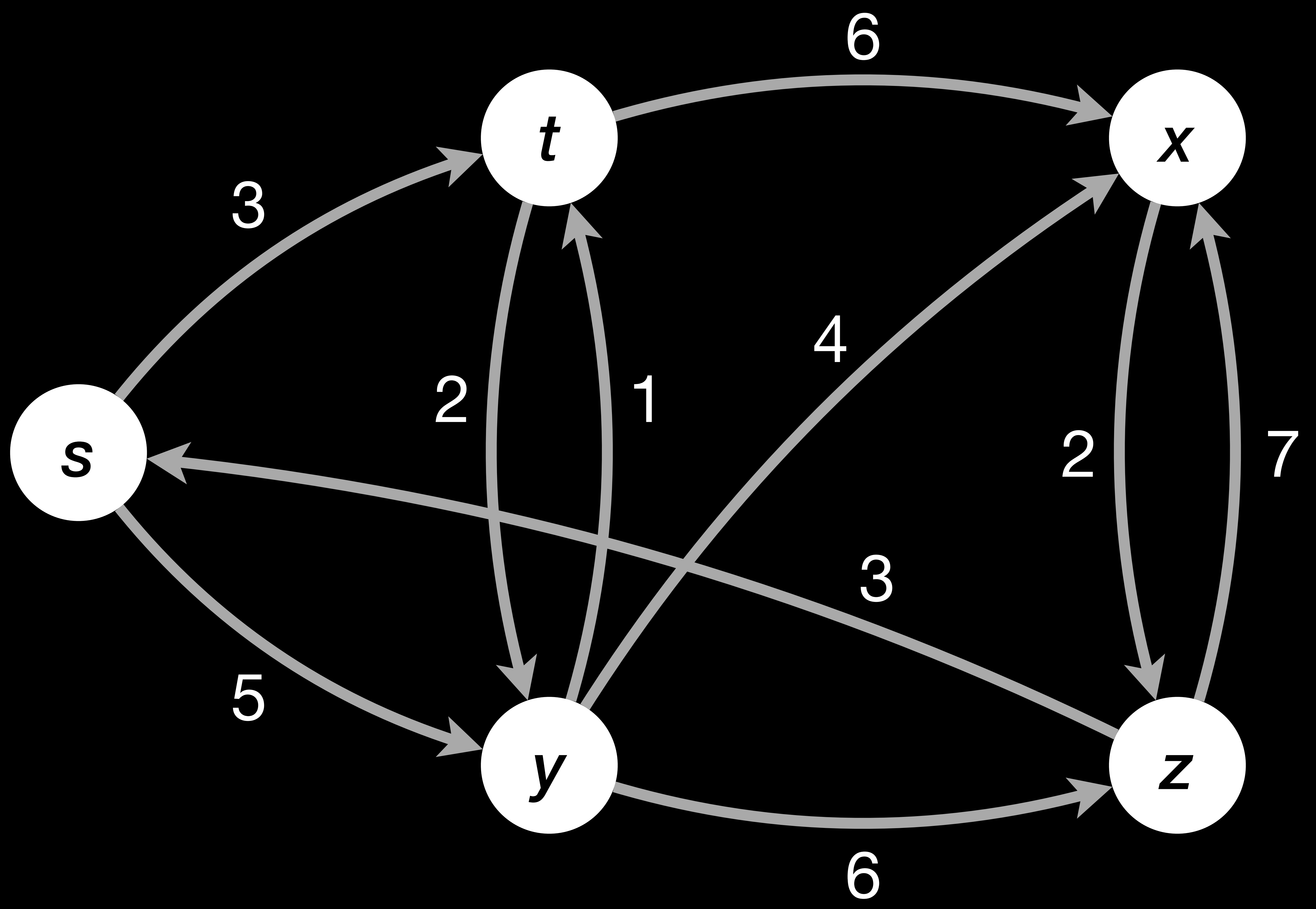
t.d = 6
6
x.d = 7
t
x
3
4
2
1
s.d = 3
s
2
7
3
5
y
z
6
y.d = ∞
z.d = 0



t.d = 6
6
x.d = 7
t
x
3
2
1
4
s.d = 3
s
2
7
3
5
y
z
y.d = 8
6
z.d = 0

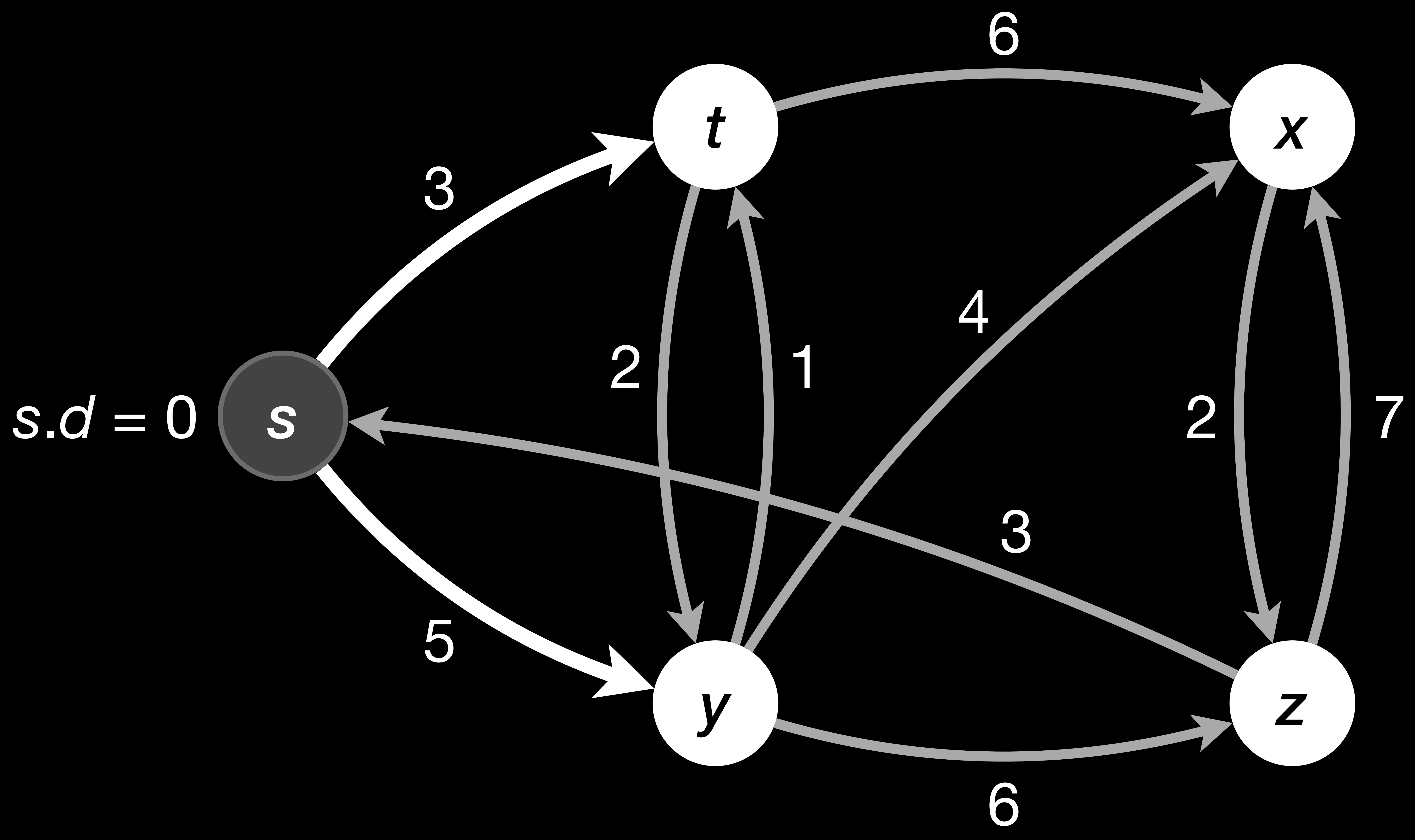
Dijkstra

source = s

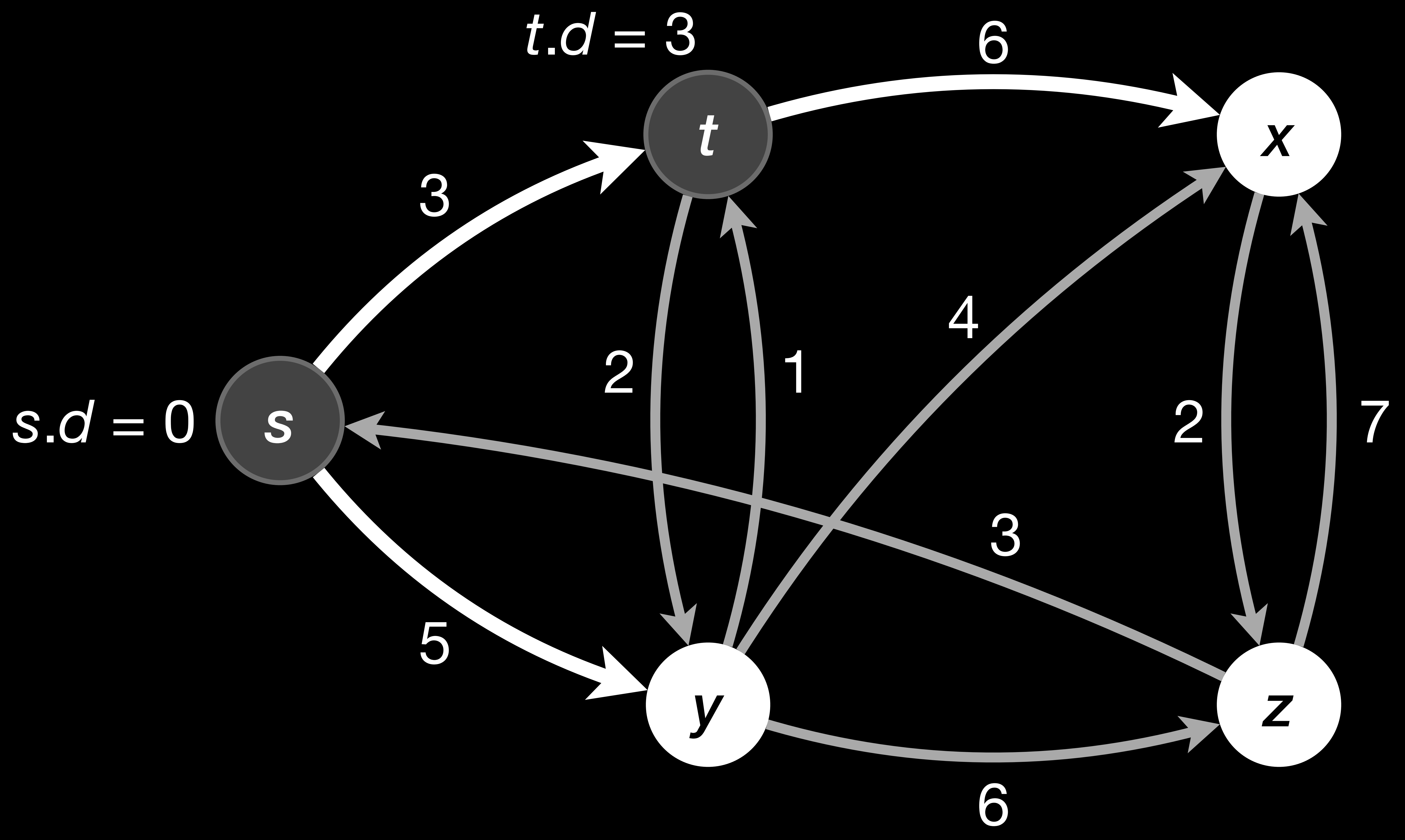


0
s

∞
t x y z

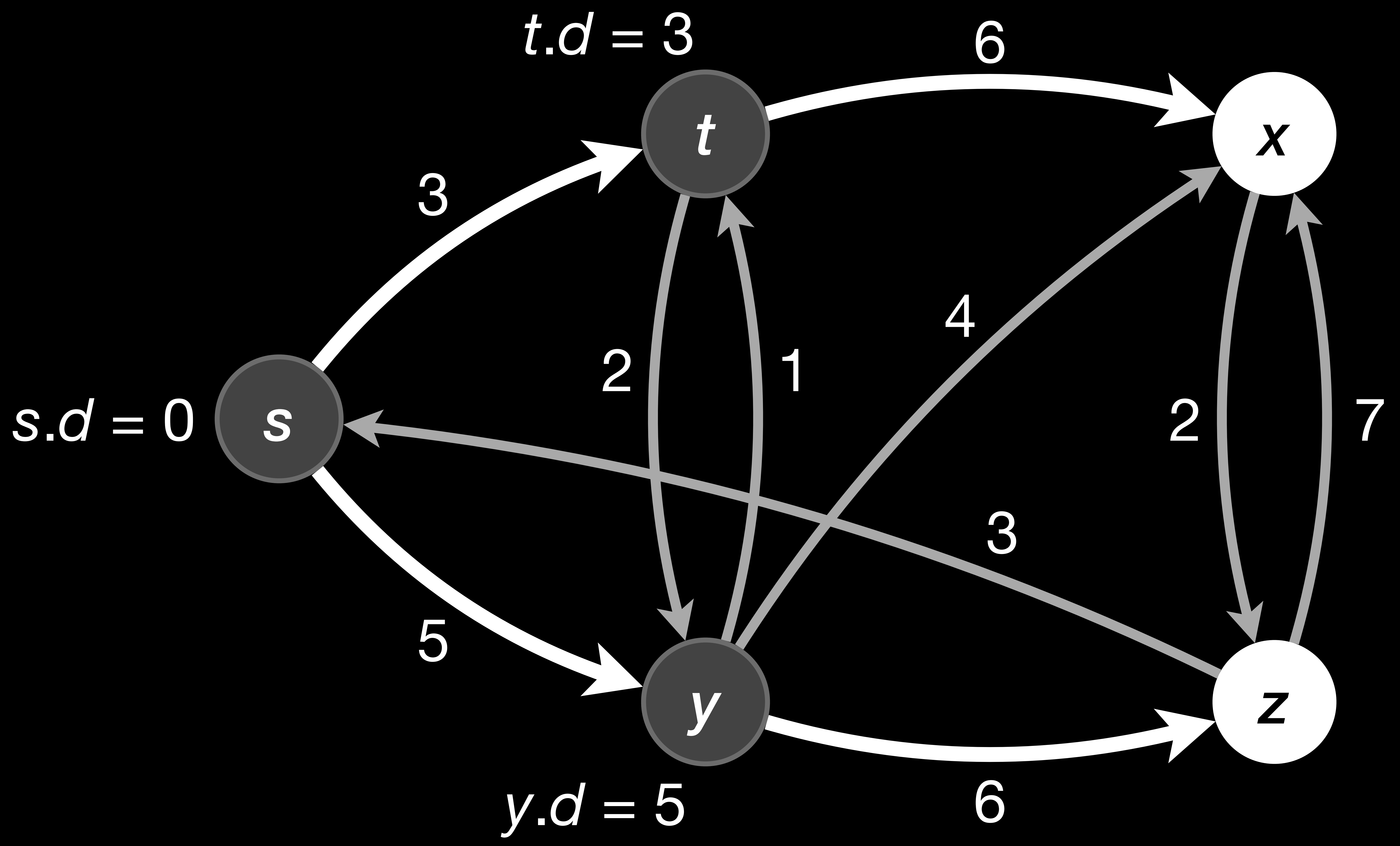


3	**5**	**∞**
t	y	x z



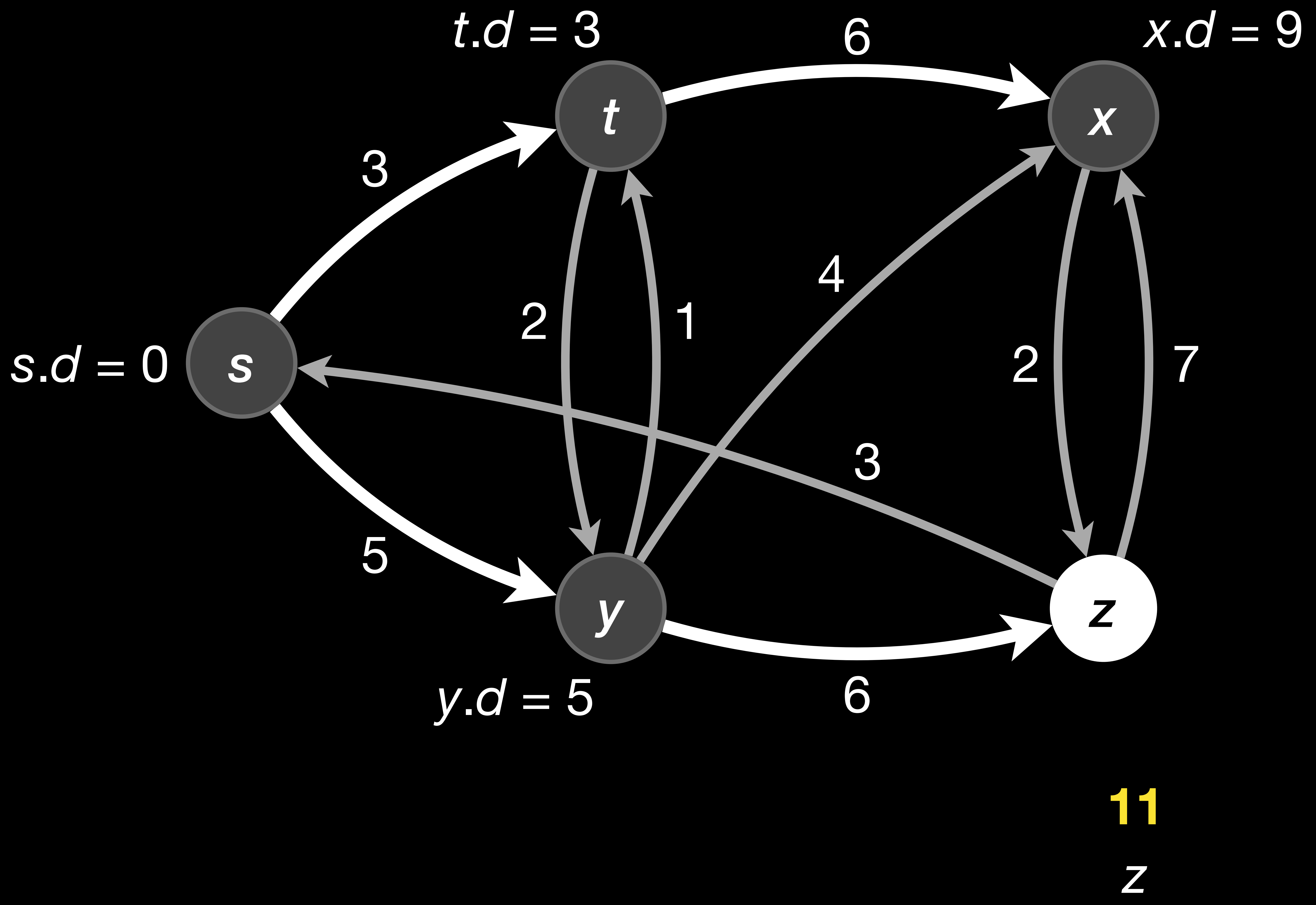
5 9 ∞

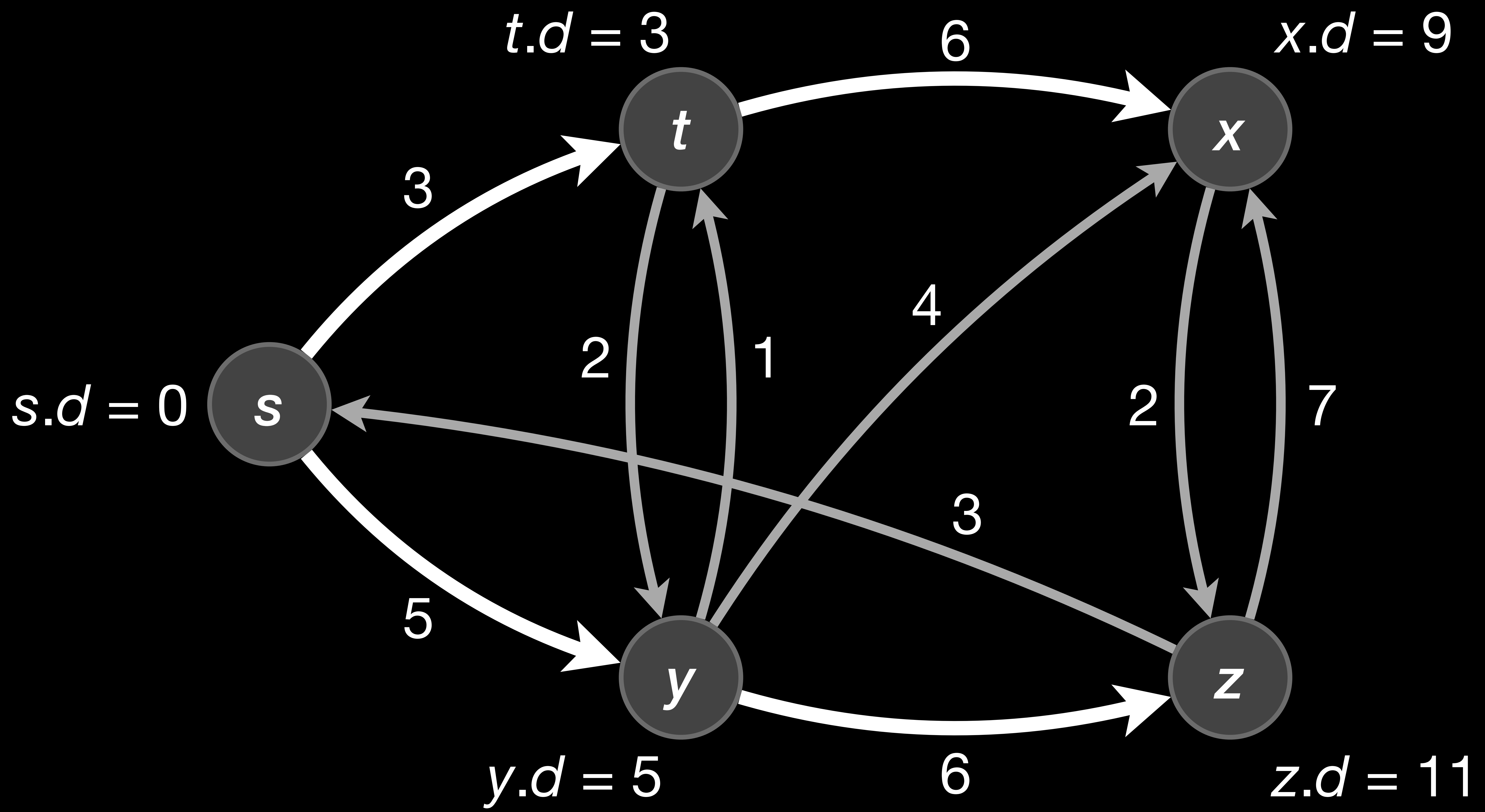
y x z



9 **11**

x z

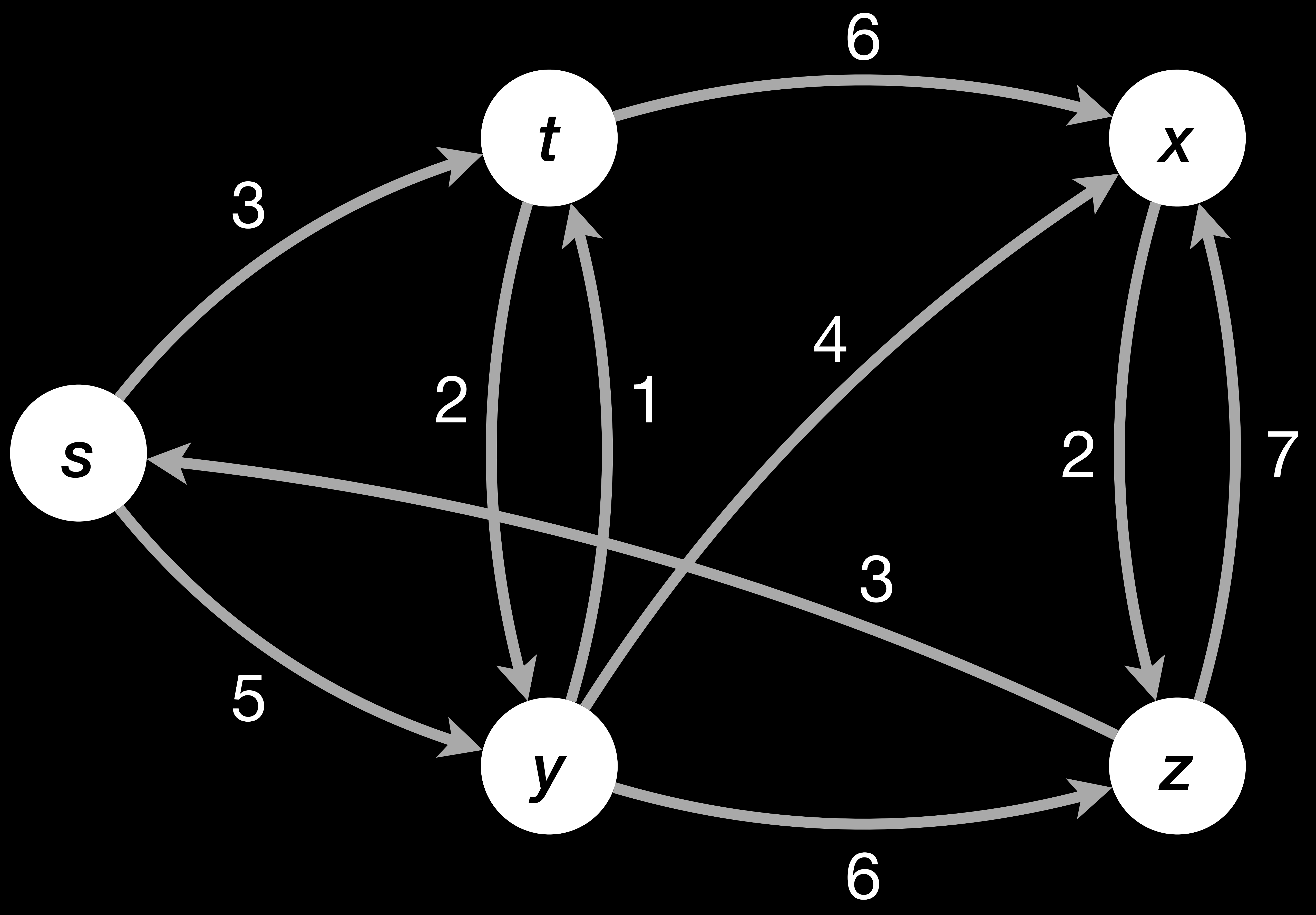

t.d = 3
6
x.d = 9
t
x
3
2
1
4
2
7
s.d = 0
s
3
5
y
z
y.d = 5
6
11
z



t.d = 3
6
x.d = 9
t
x
3
2
1
4
s.d = 0
s
2
7
3
5
y
z
y.d = 5
6
z.d = 11

Dijkstra

source = *z*

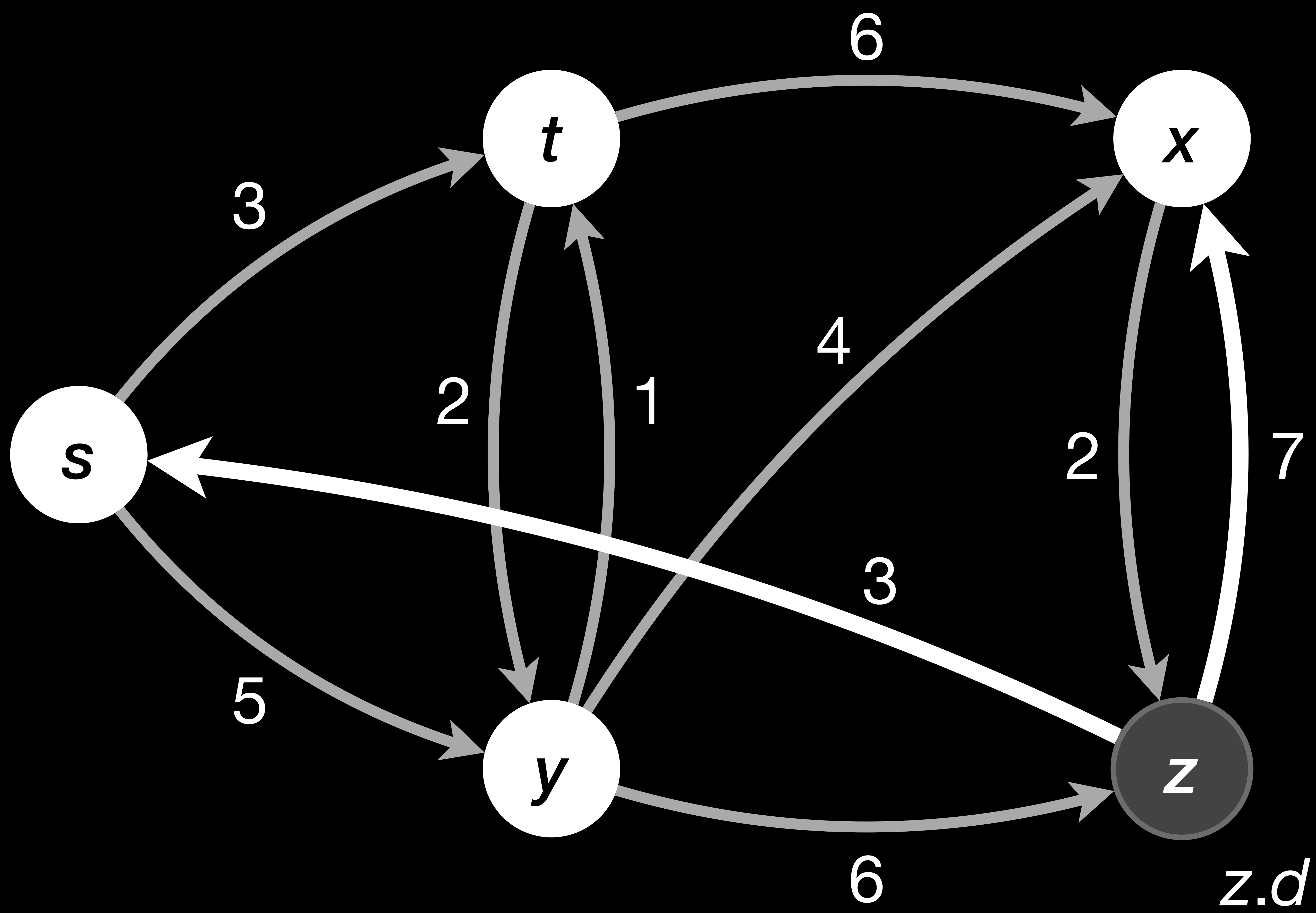


0

z

∞

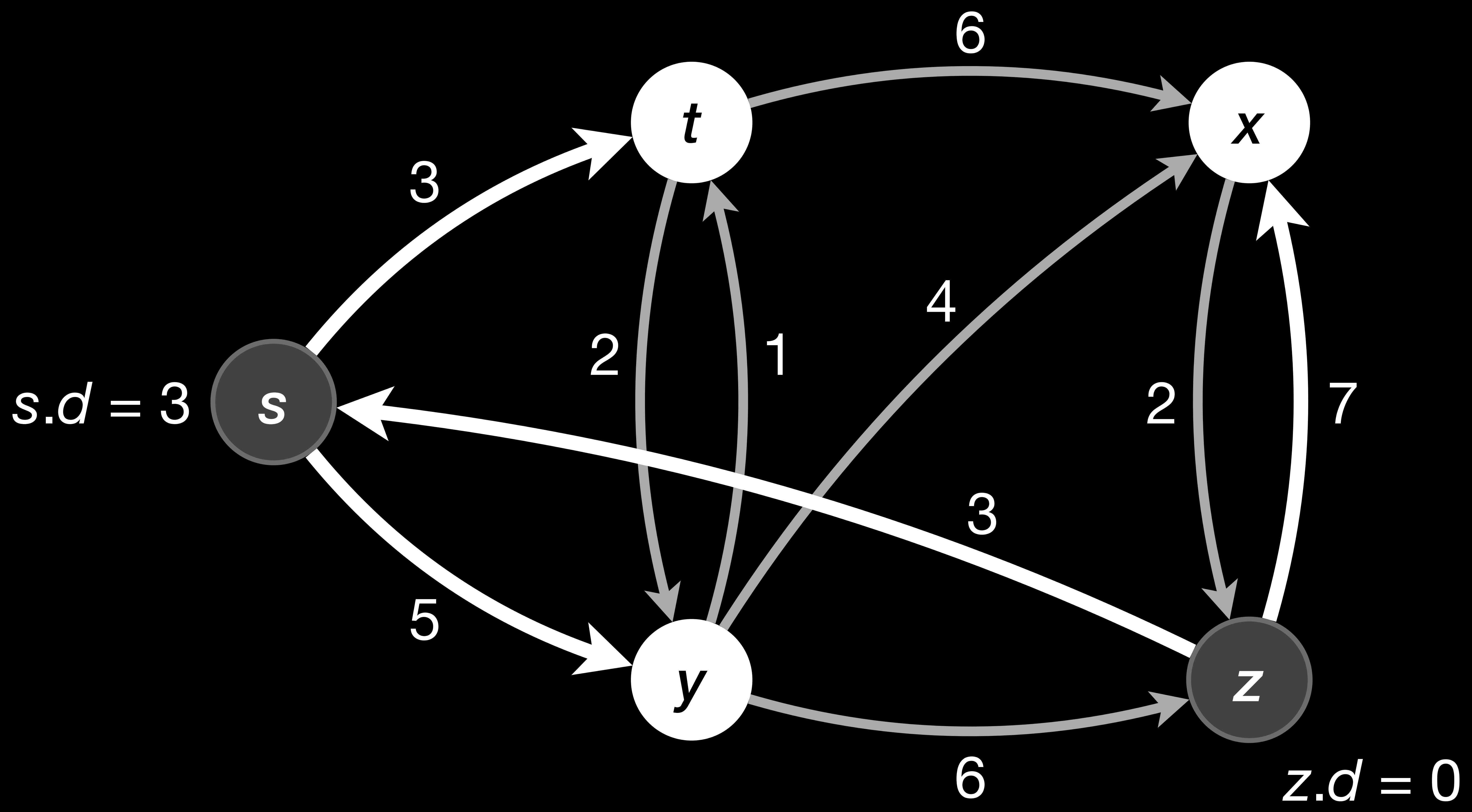
$s \quad t \quad x \quad y$



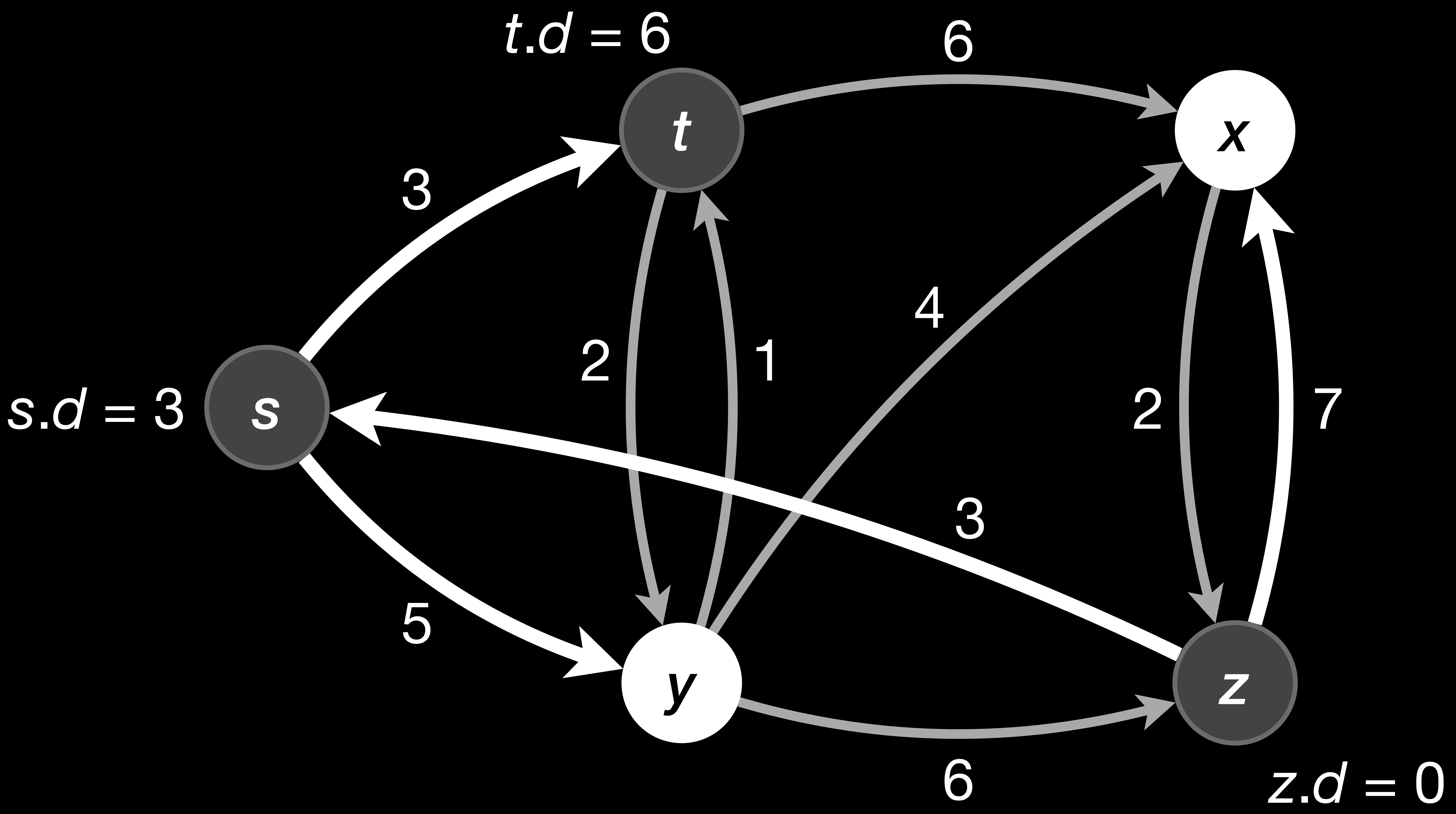
$z.d = 0$

3 **7** **∞**

s *x* *t* *y*

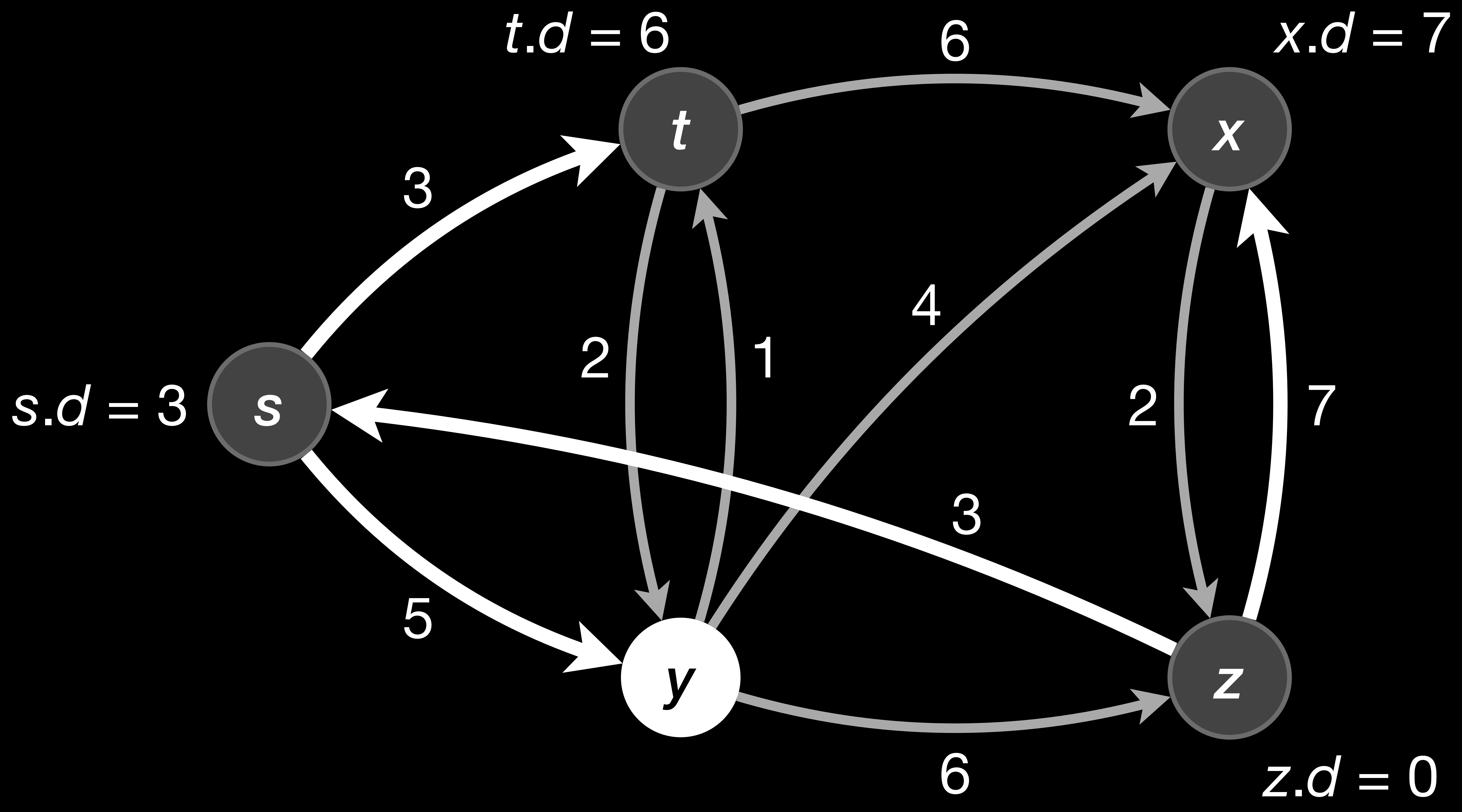


6	**7**	**8**
t	*x*	*y*



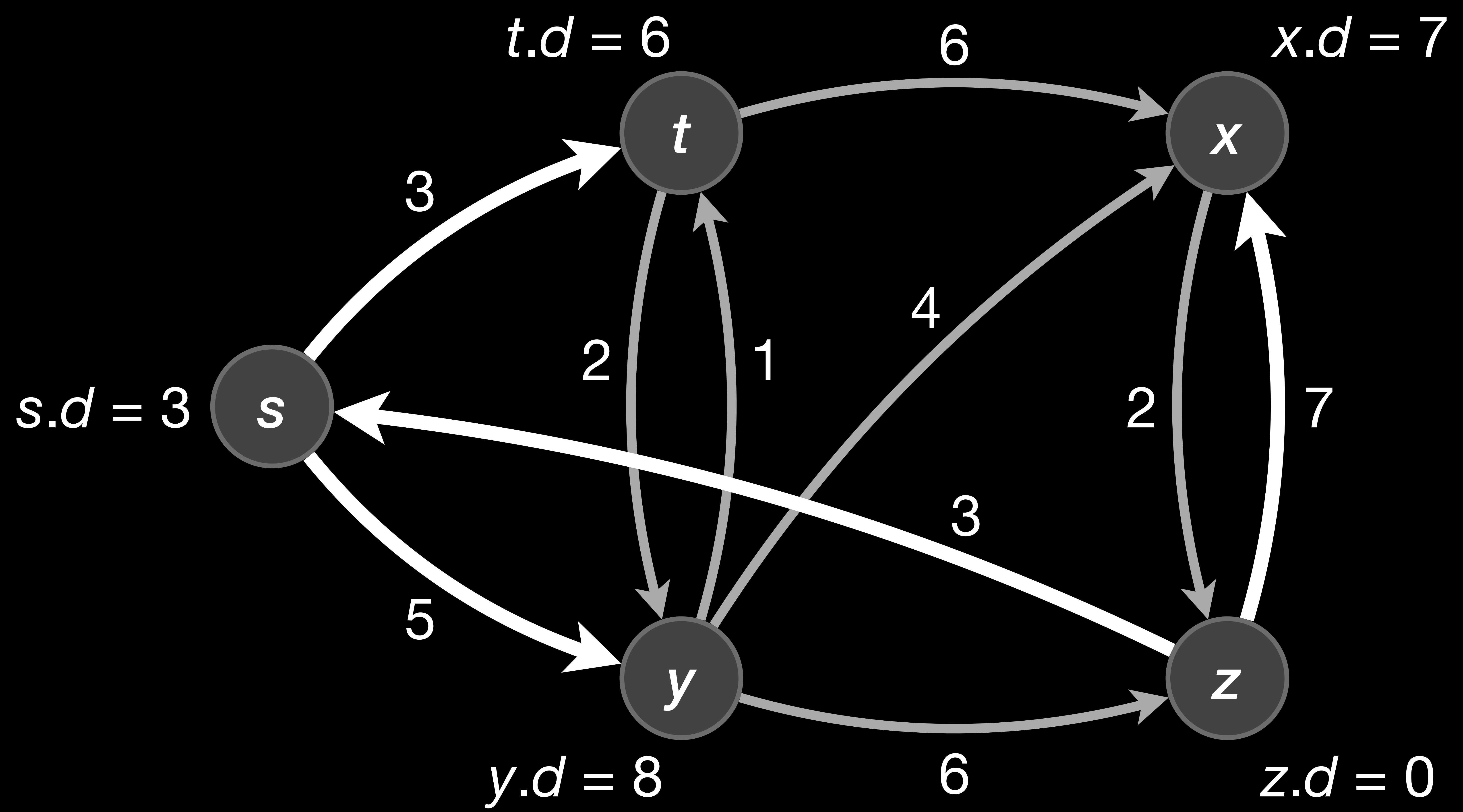
7 8

x y



8

y



t.d = 6
6
x.d = 7
t
x
3
4
2
1
2
7
s.d = 3
s
3
5
y
z
6
y.d = 8
z.d = 0