

# CSCI 432 Handout 12: Shortest Paths in Linearly Interpolated Graphs

Name: \_\_\_\_\_

2 April 2026

**Problem 1** (SP in Linearly Interpolated Graphs). *Given two weighted, directed graphs  $G_0 = (V, E, \omega_0)$  and  $G_1 = (V, E, \omega_1)$ , define the linear interpolation at parameter  $\tau \in [0, 1]$  to be*

$$G_\tau := (V, E, \omega_\tau := \tau\omega_1 + (1 - \tau)\omega_0).$$

*Given a source  $s \in V$  and a sink  $t \in V$ , we want to a data structure that will efficiently answer the query of: for  $\tau \in [0, 1]$ , what is the shortest path from  $s$  to  $t$ ?*

Recall Dijkstra's algorithm from our last class period. Here, we provide pseudocode for Dijkstra's SP algorithm for easy reference (this computes just the distance; can you adjust this to find the path too?):<sup>1</sup>

---

**Algorithm 1** Dijkstra's Shortest-Path Algorithm

---

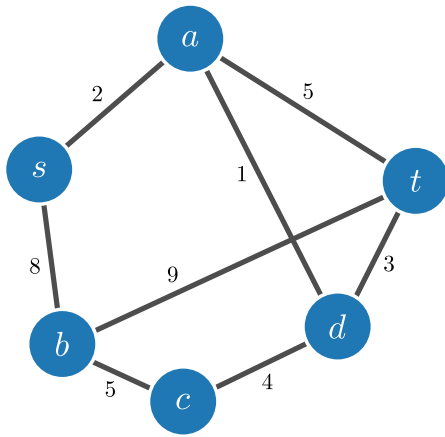
**Input:** A weighted, directed graph  $G = (V, E, \omega)$  with non-negative edge weights; a source vertex  $s \in V$

**Output:** Shortest-path distances  $\text{dist}(v)$  for all  $v \in V$

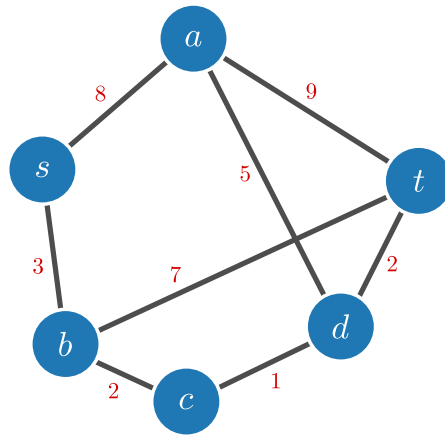
```
1: Initialize  $\text{dist}$ , a data structure indexed by  $V$ , with value  $\infty$  for all vertices except  $\text{dist}(s) = 0$ 
2:  $Q$  min-priority queue storing the elements of  $V$ , with  $\text{dist}[v]$  the priority of  $v$  for all  $v$ 
3: while  $Q \neq \emptyset$  do
4:    $u \leftarrow Q.\text{pop}()$ 
5:   for each  $(u, v)$  out of  $u$  do
6:      $alt \leftarrow \text{dist}(u) + \omega(u, v)$ 
7:     if  $alt < \text{dist}(v)$  then
8:        $\text{dist}[v] \leftarrow alt$ 
9:        $Q.\text{DECREASE-KEY}(v, \text{dist}[v])$ 
10:    end if
11:  end for
12: end while
13: return  $\text{dist}$ 
```

---

<sup>1</sup>Initial pseudocode was generated from CatChat, using the prompt "simple pseudocode for dijkstra in latex using the algorithmic package (not algpseudocode)." The output was far from perfect, so what you see here is an edited version of that output.



(a)  $G_0$



(b)  $G_1$

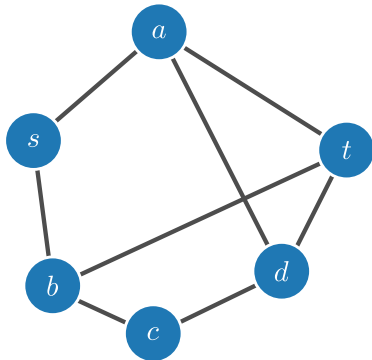
Figure 1: The starting and ending graphs that we linearly interpolate between. Here, let all edges be directed from left to right.

## Understanding the Problem

1. For  $G_0$  and  $G_1$ , use Dijkstra's algorithm to compute a shortest path from  $s$  to  $t$  in each graph (denote them by  $p_0$  and  $p_1$ , respectively). What are their lengths (denote them by  $\ell_0$  and  $\ell_1$ , respectively).

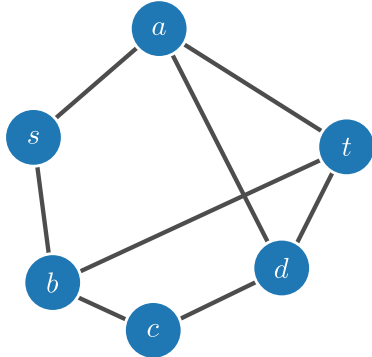
**Answer**

2. Compute  $G_{0.5}$  and find a shortest path from  $s$  to  $t$ . What is its length? (Denote the path and length by  $p_{0.5}$  and  $\ell_{0.5}$ ).



**Answer**

3. Compute  $G_{0.25}$  and find the length of a shortest path from  $s$  to  $t$ . What is its length?



**Answer**

4. For which value of  $t$  are the lengths of  $p_0$  and  $p_1$  the same (Note: there is one value  $t$  for which this holds; we denote this by  $t_{same}$ )?

**Answer**

5. What are the values of all paths between  $s$  and  $t$  at  $t_{same}$ . Is there any time that the shortest path is not  $p_0$  or  $p_1$ ? (Hint: look at the graphs of path length versus parameter  $\tau$  for all paths from  $s$  to  $t$ ).

**Answer**

6. Provide an example of two graphs for which there exists a  $t$  for which  $p_t \notin \{p_0, p_1\}$ .

**Answer**

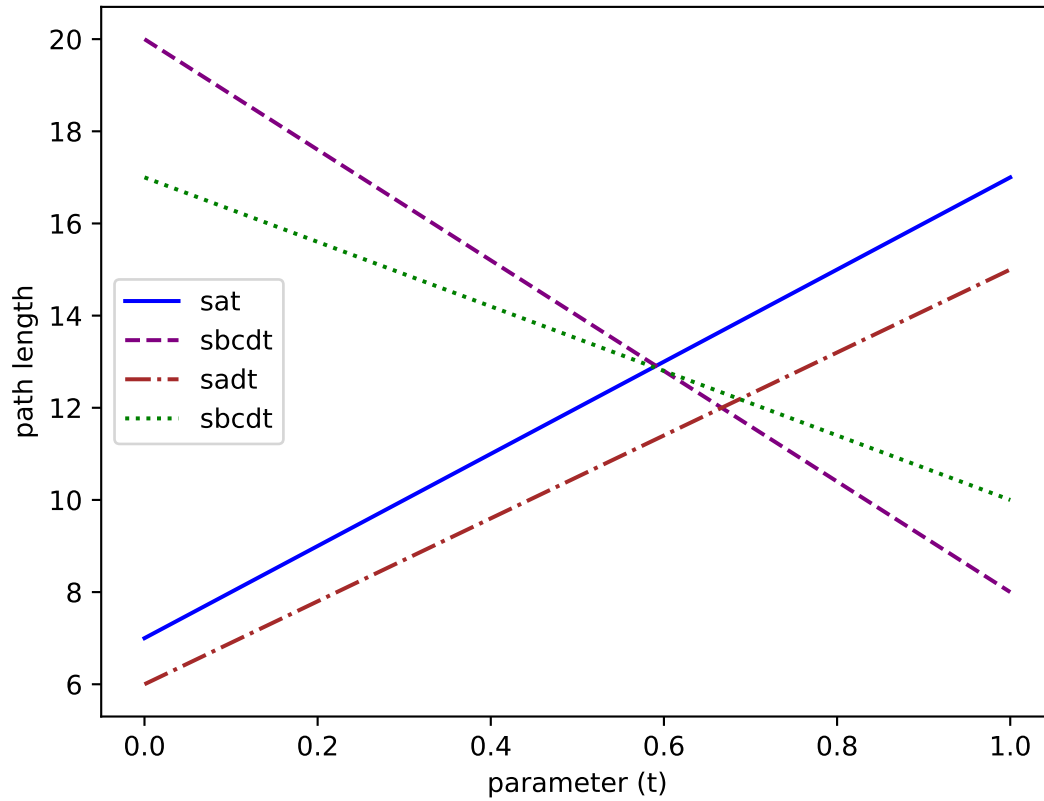


Figure 2: Path lengths.

## The Algorithm

1. What are some observations we have about the plot above? When are the shortest paths not unique?

**Answer**

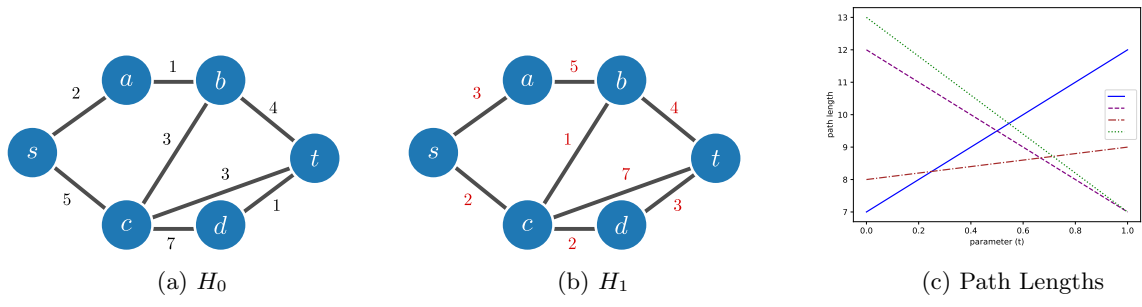


Figure 3: The starting and ending graphs that we linearly interpolate between. Here, let all edges be directed from left to right.

2. Consider the linearly interpolated graph in Figure 3, along with the graph of the lengths of all paths from  $s$  to  $t$ . What are the shortest path representatives for each parameter  $\tau$ ?

**Answer**