1 Shortest Path

1.1 s-t Shortest Path

Using the following graph as an example. Given that s is the starting node and t is the target node. Find the s-t shortest path.

The possible paths from s to t are:

1. $s \xrightarrow{5} a \xrightarrow{10} t$. The total cost is 15.
2. $s \xrightarrow{5} a \xrightarrow{1} b \xrightarrow{5} t$. The total cost is 11.
3. $s \xrightarrow{5} a \xrightarrow{1} b \xrightarrow{1} c \xrightarrow{3} t$. The total cost is 10.

1.2 Single Source Shortest Path

Problem Statement: Find the shortest path from a single source s to every other vertex in the graph.
State: Let $d[v]$ be the length of shortest path from $s$ to $v$.

Transition Function

$$d[v] = \min_{(u,v) \in E}(w(u,v) + d[u])$$

In which $w(u,v)$ is the length of last edge and $d[u]$ is the distance from $s$ to $u$.

Take the graph above as an example.

$$d[t] = \min \begin{cases} 
  d[a] + 10, & 15 \\
  d[b] + 15, & 11 \\
  d[c] + 3, & 10 
\end{cases}$$

### 1.3 Dijkstra’s Algorithm

Maintain a set of visited vertices (the vertices that we have computed shortest path for) $V$. The set is initialized as $s$, which contains only the source node. We also need to maintain a distance array.

1. For vertices that are visited. ($u \in V$).

   $$\text{dis}[u] = d[u] = \text{Length of shortest path from } s \text{ to } u.$$  

2. For vertices not visited ($u \not\in V$)

   $$\text{dis}[u] = d[u] = \text{Length of the shortest path from } s \text{ to } u, \text{ only use vertices in } V \text{ as intermediate vertices.}$$

At every iteration, select $u \not\in V$ such that dis[u] is smallest. Add $u$ to $V$, update the dis array.

**Proof of Correctness**

The main step here is to prove the claim that for vertex $v$ with smallest dis[v] among the vertices not in set $V$, $d[v] = \text{dis}[v]$.

Assume towards contradiction that there is a path from $s$ to $v$ with length shorter than dis[v]. By the inductive hypothesis, the shorter path must use vertices that are not visited as intermediate vertices. Let $v'$ be the first vertex on the path such that $v' \not\in V$. By induction hypothesis, we know distance from $s$ to $v'$ is at least dis[v'], but dis[v]dis[v'] by choice of the algorithm, so length of this path cannot be smaller than dis[v].