SINGLE SOURCE  SHORTEST PATHS

paths from $s$ to $t$

$3 + 2 + 1 + 20$
SINGLE SOURCE SHORTEST PATHS

paths from \( s \) to \( t \)

\[ 3 + 2 + 1 + 20 \]

\[ 3 + 2 + 4 + 15 \]

not greedy or BFS

although this is an extension of BFS (weights = 1)
SINGLE SOURCE SHORTEST PATHS

paths from $s$ to $t$

$3 + 2 + 1 + 20$

$3 + 2 + 4 + 15$

$3 + 2 + 4 + 5 + 8 = 22$

not greedy or BFS

tot. length of path from $s$
SINGLE SOURCE SHORTEST PATHS

paths from s to t

\[ 3 + 2 + 1 + 20 \]
\[ 3 + 2 + 4 + 15 \]
\[ 3 + 2 + 4 + 5 + 8 = 22 \]

multiple options

\[ \frac{5}{5} + 1 + \cdots \]
\[ + 4 + \cdots = 22 \]

Generally assume a directed graph (can make undirected→directed easily)

disconnected from s

not greedy or BFS
SINGLE SOURCE SHORTEST PATHS

Observations
- No cycles in $s \rightarrow t$ (shortest path)

assumption?
SINGLE SOURCE SHORTEST PATHS

Observations
- No cycles in \( s \rightarrow t \)
- Negative weights \( \sim \) OK, unless they form a negative cycle in \( G \)

Any vertex reachable from a negative cycle gets a score of \(-\infty\)

\( \{ \) assuming cycle can be reached from \( s \) \( \} \)
SINGLE SOURCE SHORTEST PATHS

Observations
- No cycles in $s \rightarrow t$
- Negative weights are OK, unless they form a negative cycle in $G$

Any vertex reachable from a negative cycle gets a score of $-\infty$
**Observations**

- No cycles in $s \rightarrow t$
- Negative weights are OK, unless they form a negative cycle in $G$
- Shortest path $s \rightarrow v \rightarrow t$ contains
  - Shortest path $s \rightarrow v$ (9)
  - Shortest path $v \rightarrow t$ (13)
to any vertex $t$

there may be multiple shortest paths

e.g. $s \rightarrow t$ or $s \rightarrow x \rightarrow y \rightarrow t$
to any vertex \( t \)

there may be multiple shortest paths

\( e.g. \ s \to t \text{ or } s \to x \to y \to t \)

All shortest paths from \( s \) to \( V \) can be represented in a DAG

\( \text{DAG} \to \text{tree} : \text{arbitrarily keep one path to each vertex} \)

"shortest paths tree"

(similar to picking one BFS/DFS search)
By exploring some path from $s$ to $t$ we get a path score (e.g. 26) the score of $t$ is 26, which may only decrease as we explore more options.

If we find the score of $v$: $d(v)$ & $\exists$ edge $v \rightarrow t$

then we can possibly improve $d(t)$:

$$d(v) + w(v, t) < d(t)$$

(15)
By exploring some path from \( s \) to \( t \) we get a path score (e.g., 26) the score of \( t \) is 26, which may only decrease as we explore more options.

If we find the score of \( v \) \( d(v) \) & \( \exists \) edge \( v \rightarrow t \) then we can possibly improve \( d(t) \):

\[
d(v) + w(v, t) < d(t) ?
\]

Could also improve \( d(t) \) if the score of one of its ancestors improves
By exploring some path from \( s \) to \( t \) we get a path score (e.g. 26) the score of \( t \) is 26, which may only decrease as we explore more options.

If we find the score of \( v \) \( d(v) \) & \( \exists \) edge \( v \rightarrow t \) then we can possibly improve \( d(t) \):

\[
d(v) + w(v, t) < d(t)\]

Could also improve \( d(t) \) if the score of one of its ancestors improves
By exploring some path from $s$ to $t$ we get a path score (e.g. 26) the score of $t$ is 26, which may only decrease as we explore more options.

If we find the score of $v$ $d(v)$ & $\exists$ edge $v \rightarrow t$
then we can possibly improve $d(t)$:

$$d(v) + w(v, t) < d(t)$$

Could also improve $d(t)$ if the score of one of its ancestors improves.
By exploring some path from $s$ to $t$ we get a path score (e.g. 26), the score of $t$ is 26, which may only decrease as we explore more options.

If we find the score of $v$ $d(v)$ & $\exists$ edge $v \rightarrow t$ then we can possibly improve $d(t)$:

$$d(v) + w(v, t) < d(t)?$$

Could also improve $d(t)$ if the score of one of its ancestors improves.
By exploring some path from \( s \) to \( t \) we get a path score (e.g. 26) the score of \( t \) is 26, which may only decrease as we explore more options.

If we find the score of \( v \) \( d(v) \) & \( \exists \) edge \( v \to t \) then we can possibly improve \( d(t) \):

\[
d(v) + w(v, t) < d(t)\
\]

Could also improve \( d(t) \) if the score of one of its ancestors improves
Relax\((v,t)\): checking if score of \(t\) can be improved (lowered) by using \(s \rightarrow v \rightarrow t\)

Keep min of \(d(t)\) vs. \(d(v) + w(v,t)\)

If \(v\) helps, then parent\((t) = v\)
Assume this is a shortest path from $s$ to $t$ unknown but exists size: $k < V$

Suppose we have an algorithm based on relaxing edges. If we relax $e_1$ before $e_2$ before $\ldots$ before $e_{k-1}$ before $e_k$ then we will correctly compute $d(t)$.

Relax sequence: $e^* e_1 e^j e^* e_2 e^* e_1 e_k e_{k-1} e_i e^* e_k e^y$ : ok (don't care if we relax other edges or the same ones repeatedly)