Johnson's Algorithm for Sparse Graphs

Using Dijkstras gives best complexity for sparse graphs. However Dijkstra works only for positive weights.

Convert graph to contain only positive weight edges and then use Dijkstras starting at every node.

Requirements:

Preserving shortest paths after edge reweighting :

Preserving negative cycles

Producing nonnegative edge weights by reweighting :

Johnson's Algorithm for Sparse Graphs

(1) Preserving shortest paths by edge reweighting :

- L1 : given $G = (V , E)$ with $\omega : E \rightarrow R$
  - ▶ let $h : V \rightarrow R$ be any weighting function on the vertex set
  - ▶ define $\hat{\omega}(\omega , h) : E \rightarrow R$ as $\hat{\omega}(u , v) = \omega(u , v) + h (u) - h (v)$
  - ▶ let $p_{0k} = < v_0 , v_1 , \ldots , v_k >$ be a path from v_0 to v_k

(a) $\hat{\omega}(p_{0k}) = \omega(p_{0k}) + h (v_0) - h (v_k)$

(b) $\omega(p_{0k}) = \delta(v_0, v_k) \text{ in } (G, \omega) \Leftrightarrow \hat{\omega}(p_{0k}) = \hat{\delta}(v_0, v_k) \text{ in } (G, \hat{\omega})$

(c) (G , ω) has a neg-wgt cycle $\Leftrightarrow$ $(G , \hat{\omega})$ has a neg-wgt cycle

Johnson's Algorithm for Sparse Graphs

- proof (a): $\hat{\omega}(p_{0k}) = \sum_{1 \leq i \leq k} \hat{\omega}(v_{i-1}, v_i)$
  $= \sum_{1 \leq i \leq k} (\omega(v_{i-1}, v_i) + h(v_0) - h(v_k))$
  $= \sum_{1 \leq i \leq k} \omega(v_{i-1}, v_i) + \sum_{1 \leq i \leq k} (h(v_0) - h(v_k))$
  $= \omega(p_{0k}) + h(v_0) - h(v_k)$

- proof (b): ($\Rightarrow$) show $\omega(p_{0k}) = \delta(v_0, v_k) \Rightarrow \hat{\omega}(p_{0k}) = \hat{\delta}(v_0, v_k)$ by contradiction.
  - ► Suppose that a shorter path p_{0k}' from v_0 to v_k in $(G, \hat{\omega})$, then $\hat{\omega}(p_{0k}') < \hat{\omega}(p_{0k})$

- due to (a) we have
  - $\omega(p_{0k}') + h(v_0) - h(v_k) = \hat{\omega}(p_{0k}') < \hat{\omega}(p_{0k}) = \omega(p_{0k}) + h(v_0) - h(v_k)$
    $\omega(p_{0k}') + h(v_0) - h(v_k) < \omega(p_{0k}) + h(v_0) - h(v_k)$
    $\omega(p_{0k}') < \omega(p_{0k}) \Rightarrow$ contradicts that p_{0k} is a shortest path in (G, ω)

Johnson's Algorithm for Sparse Graphs

- proof (b): (<=) similar

- proof (c): ($\Leftrightarrow$) consider a cycle $c = < v_0 , v_1 , \ldots , v_k = v_0 >$.

  Due to (a)

  ► $\hat{\omega}(c) = \sum_{1 \le i \le k} \hat{\omega}(v_{i-1}, v_i) = \omega(c) + h(v_0) - h(v_k)$

  $= \omega(c) + h(v_0) - h(v_0) = \omega(c)$ since $v_k = v_0$

  ► $\hat{\omega}(c) = \omega(c)$.

QED

Johnson's Algorithm for Sparse Graphs

(2) Producing nonnegative edge weights by reweighting :

- given (G, ω) with $G = (V, E)$ and $\omega : E \rightarrow R$
  construct a new graph (G', ω') with $G' = (V', E')$ and
  $\omega' = E' \rightarrow R$
  - ▶ $V' = V \cup \{ s \}$ for some new vertex $s \notin V$
  - ▶ $E' = E \cup \{ (s, v) : v \in V \}$
  - ▶ $\omega'(u,v) = \omega(u,v) \quad \forall (u,v) \in E$ and $\omega'(s,v) = 0, \quad \forall v \in V$

- vertex s has no incoming edges $\Rightarrow s \notin R_v$ for any v in V
  - ▶ no shortest paths from $u \neq s$ to v in G' contains vertex s
  - ▶ (G', ω') has no neg-wgt cycle $\Leftrightarrow$ (G, ω) has no neg-wgt cycle

Johnson's Algorithm for Sparse Graphs

- suppose that G and G' have no neg-wgt cycle

- L2 : if we define $h(v) = \delta(s, v)$ $\forall v \in V$ in G' and $\hat{\omega}$ according to L1.

  ► we will have $\hat{\omega}(u,v) = \omega(u,v) + h(u) - h(v) \geq 0 \quad \forall v \in V$

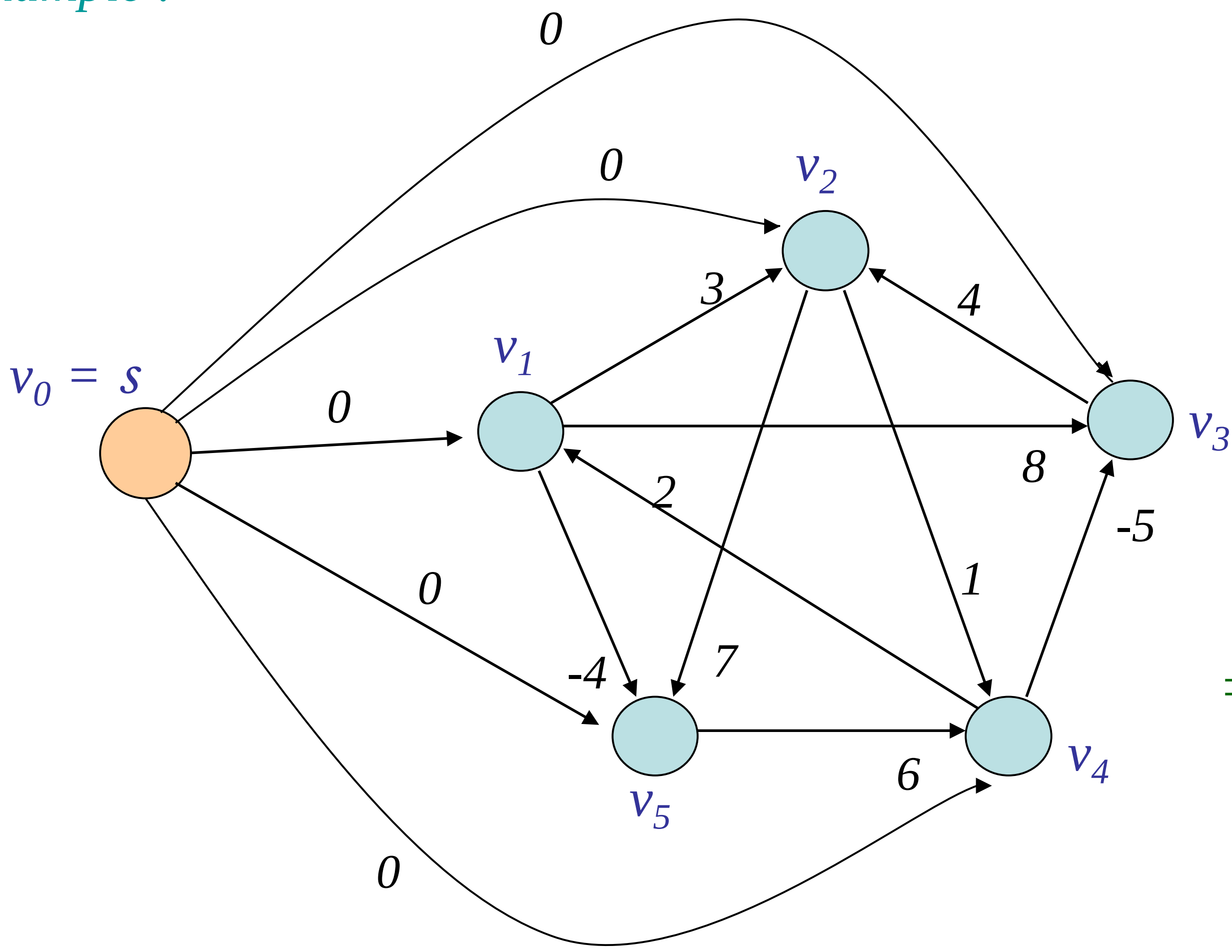
proof : for every edge $(u, v) \in E$

$$\delta(s, v) \leq \delta(s, u) + \omega(u, v) \text{ in } G' \text{ due to triangle inequality}$$

$$h(v) \leq h(u) + \omega(u, v) \Rightarrow 0 \leq \omega(u, v) + h(u) - h(v) = \hat{\omega}(u, v)$$

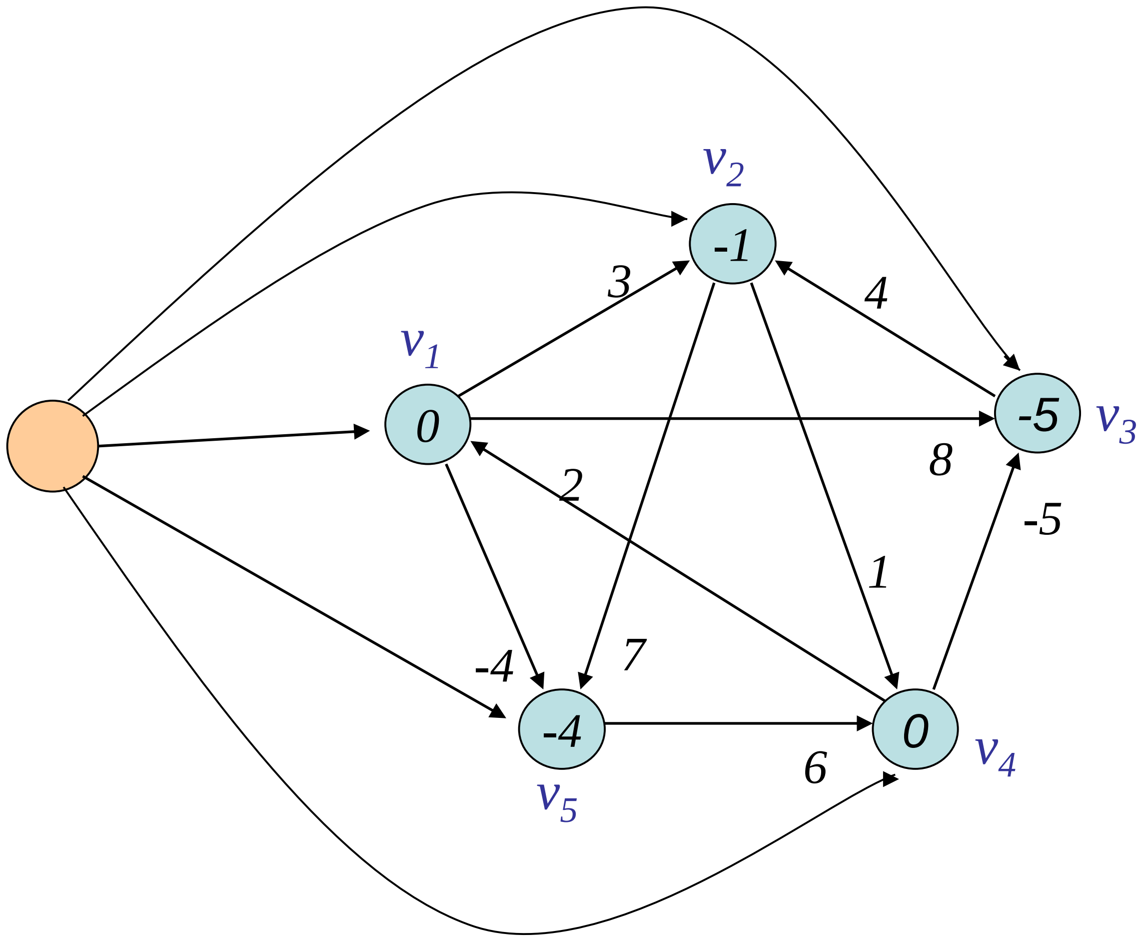
Johnson's Algorithm for Sparse Graphs

example :



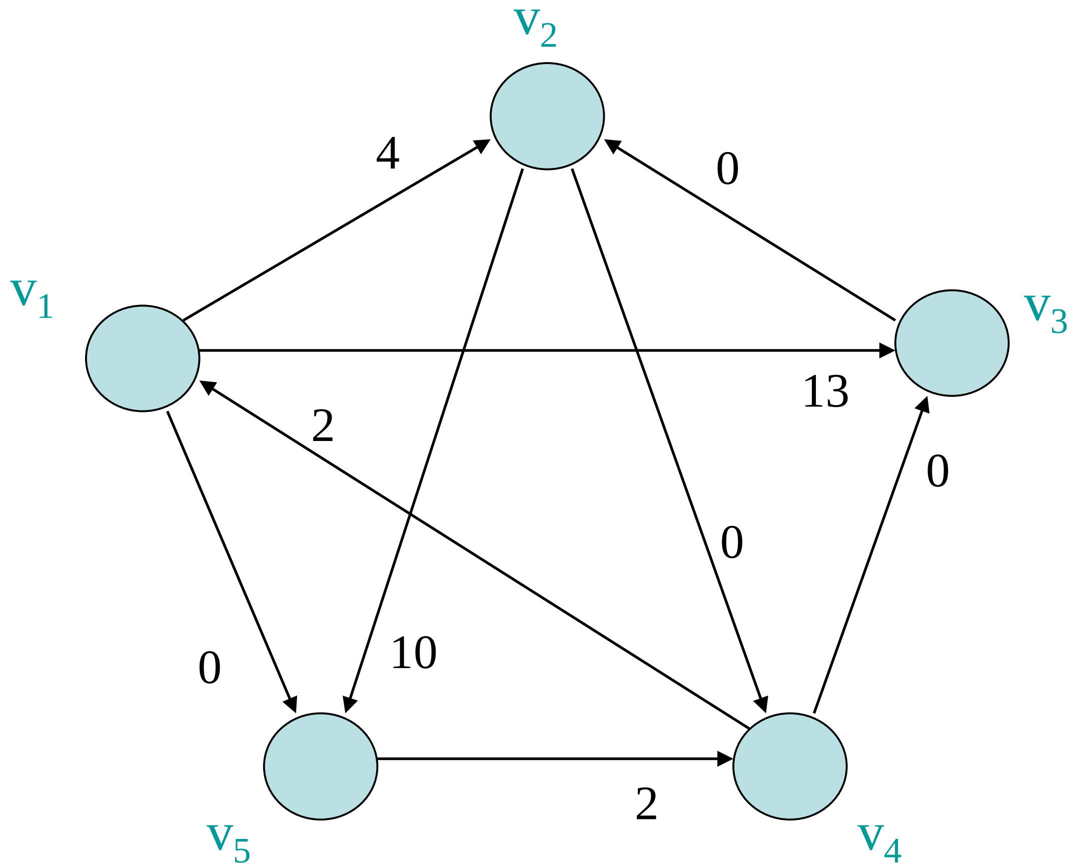
$\Rightarrow (\,G', \omega'\,)$

Johnson’s Algorithm for Sparse Graphs
Edge Reweighting



$\Rightarrow (G', \omega')$ with $h(v)$

Johnson's Algorithm for Sparse Graphs
Edge Reweighting



$\Rightarrow (G, \hat{\omega})$

Johnson's Algorithm for Sparse Graphs

Computing All-Pairs Shortest Paths

- adjacency list representation of G.

- returns n x n matrix $D = (d_{ij})$ where

  $d_{ij} = \delta_{ij}$,

  or reports the existence of a neg-wgt cycle.

Johnson's Algorithm for Sparse Graphs

- JOHNSON(G,ω)

  ► D=(d_{ij}) is an nxn matrix

  ► construct ($G' = (V', E')$, ω') s.t. $V' = V \cup \{s\}$; $E' = E \cup \{ (s,v) : \forall v \in V \}$

  ► $\omega'(u,v) = \omega(u,v), \quad \forall (u,v) \in E \quad \& \quad \omega'(s,v) = 0 \quad \forall v \in V$

  if BELLMAN-FORD(G', ω', s) = FALSE then

  return "negative-weight cycle"

  else

  for each vertex $v \in V' - \{s\} = V$ do

  $h[v] \leftarrow d'[v]$ ► $d'[v] = \delta'(s,v)$ computed by BELLMAN-FORD(G', ω', s)

  for each edge $(u,v) \in E$ do

  $\hat{\omega}(u,v) \leftarrow \omega(u,v) + h[u] - h[v]$ ► edge reweighting

  for each vertex $u \in V$ do

  run DIJKSTRA(G, $\hat{\omega}$, u) to compute $\hat{d}[v] = \hat{\delta}(u,v)$ for all v in $V \in (G,\hat{\omega})$

  for each vertex $v \in V$ do

  $d_{uv} = \hat{d}[v] - (h[u] - h[v])$

  return D

Johnson's Algorithm for Sparse Graphs

- running time : $O(V^2 \lg V + EV)$

  ► edge reweighting

  BELLMAN-FORD(G', ω', s) : $O(EV)$

  computing $\hat{\omega}$ values : $O(E)$

  ► $|V|$ runs of DIJKSTRA : $|V| \times O(V \lg V + EV)$

  $= O(V^2 \lg V + EV)$;

  PQ = fibonacci heap