# All-Pairs Shortest Paths

# All-Pairs Shortest Paths

- Solve $\delta(u,v)$ for all $u,v \in V$.
- Run Bellman-Ford for every $v \in V$:
  - $O(|V|^2|E|)=O(|V|^4)$
- Run Dijkstra's algorithm for every $v \in V$:
  - $O(|V|^3)$ Array
  - $O(|V||E|\log|V|)$ Binary heap
  - $O(|V|^2\log|V|+|V||E|)$ Fibonacci heap

# All-Pairs Shortest Paths

- In Chapter 25, the textbook gives several algorithms solving APSP for graphs without negative cycles.
- DP: $O(|V|^4)$
- DP+Fast Exponentiation: $O(|V|^3 \log|V|)$
- Floyd-Warshall: $O(|V|^3)$
- Johnson's: Bellman-Ford+$|V| \times$Dijkstra's
  - Negative edges

# Floyd-Warshall

- $G=(V,E)$ where $V=\{v_1,\ldots,v_n\}$
- Dynamic programming
- Subproblem:
  - $D^i(u,v)$ is the minimum length of paths from u to v which only pass vertices in $\{v_1,\ldots,v_i\}$.
  - $D^i(v,v)=0$
  - $D^0(u,v)=w(u,v)$ $w(u,v)=\infty$ if $u\neq v$ and $(u,v)\notin E$
  - $D^i(u,v)=\min(D^{i-1}(u,v),D^{i-1}(u,v_i)+D^{i-1}(v_i,v))$
  - $D^n(u,v)=\delta(u,v)$

# Floyd-Warshall

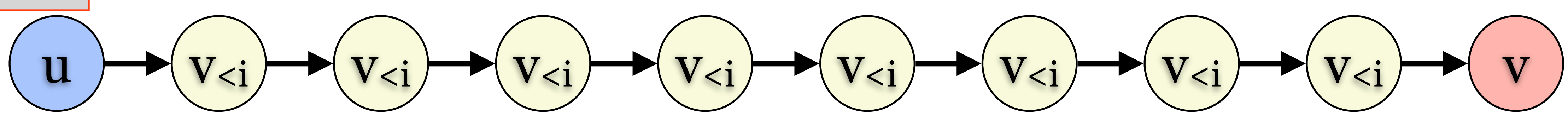
- $D^0 = W$
  for $i = 1$ to $n$
      for $u \in V$
          for $v \in V$
              $D^i(u,v) = \min(D^{i-1}(u,v), D^{i-1}(u,v_i) + D^{i-1}(v_i,v))$
  return $D^n$
- Predecessor
  - $\Pi^0(u,v) = u$ if $u \neq v$ and $(u,v) \in E$.
  - $\Pi^0(u,v) = NIL$ if $u = v$ or $(u,v) \notin E$.
  - $\Pi^i(u,v) = \Pi^{i-1}(u,v)$ if $D^i(u,v) = D^{i-1}(u,v)$
  - $\Pi^i(u,v) = \Pi^{i-1}(v_i,v)$ if $D^i(u,v) \neq D^{i-1}(u,v)$

# Correctness

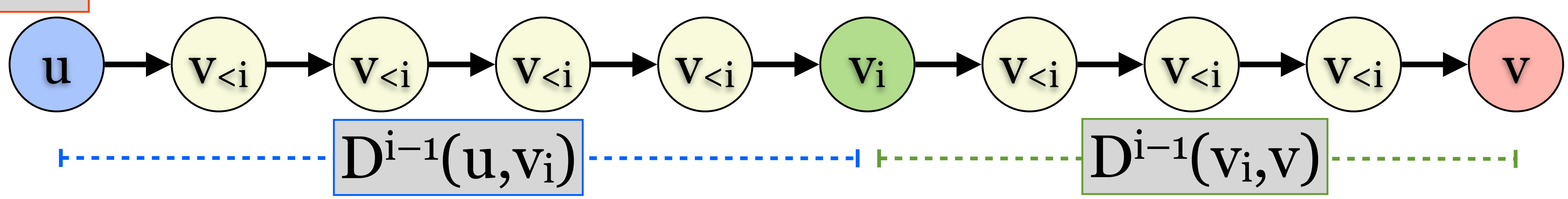
p does not contain a cycle.

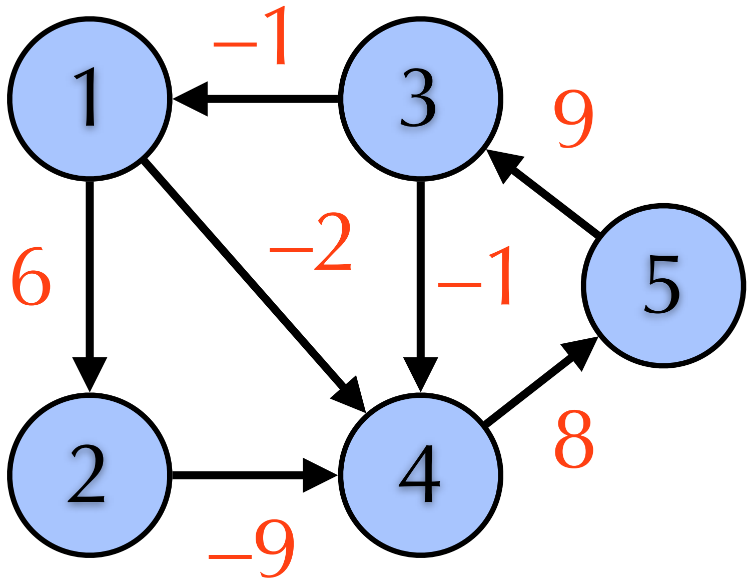
- Let p be the shortest path from u to v which only pass vertices in $\{v_1,\ldots,v_i\}$. $w(p)=D^i(u,v)$
  - Case 1: p does not pass $v_i$. So $w(p)=D^{i-1}(u,v)$.
  - Case 2: p passes $v_i$. $w(p)=D^{i-1}(u,v_i)+D^{i-1}(v_i,v)$.

Case 1




Case 2

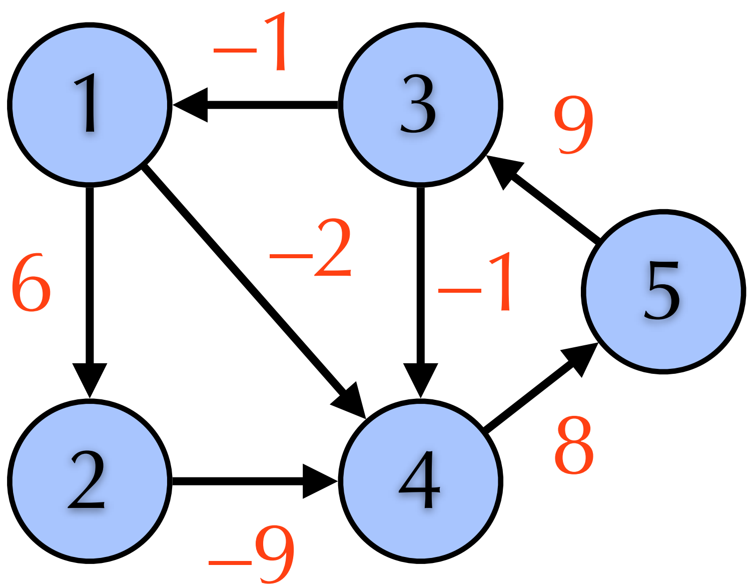





# Example

| $D^0$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 6 | ∞ | –2 | ∞ |
| 2 | ∞ | 0 | ∞ | –9 | ∞ |
| 3 | –1 | ∞ | 0 | –1 | ∞ |
| 4 | ∞ | ∞ | ∞ | 0 | 8 |
| 5 | ∞ | ∞ | 9 | ∞ | 0 |

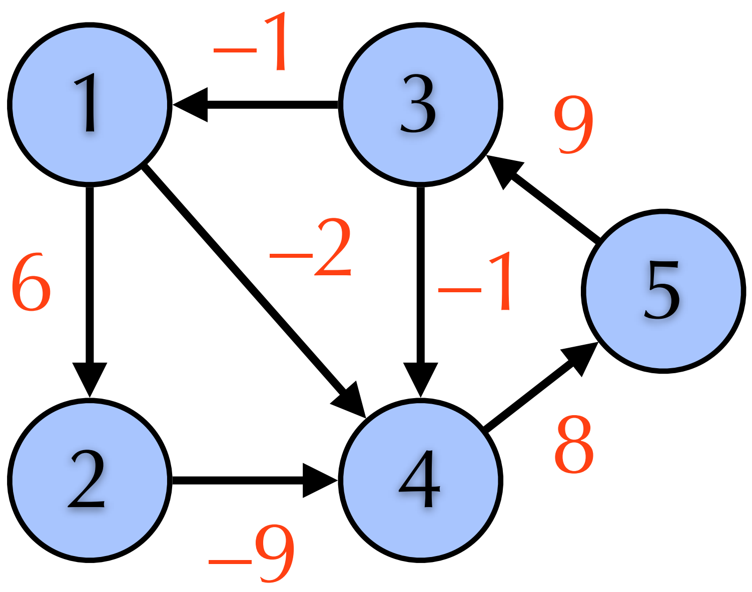
| $\Pi^0$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | NIL | 1 | NIL | 1 | NIL |
| 2 | NIL | NIL | NIL | 2 | NIL |
| 3 | 3 | NIL | NIL | 3 | NIL |
| 4 | NIL | NIL | NIL | NIL | 4 |
| 5 | NIL | NIL | 5 | NIL | NIL |




# Example

| $D^1$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 6 | ∞ | –2 | ∞ |
| 2 | ∞ | 0 | ∞ | –9 | ∞ |
| 3 | –1 | **5** | 0 | **–3** | ∞ |
| 4 | ∞ | ∞ | ∞ | 0 | 8 |
| 5 | ∞ | ∞ | 9 | ∞ | 0 |

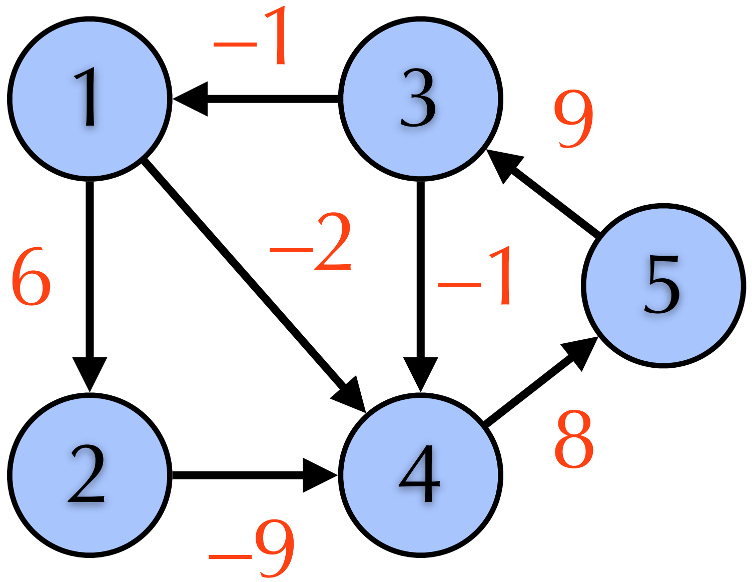
| $\Pi^1$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | NIL | 1 | NIL | 1 | NIL |
| 2 | NIL | NIL | NIL | 2 | NIL |
| 3 | 3 | **1** | NIL | **1** | NIL |
| 4 | NIL | NIL | NIL | NIL | 4 |
| 5 | NIL | NIL | 5 | NIL | NIL |




# Example

| $D^2$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 6 | ∞ | **–3** | ∞ |
| 2 | ∞ | 0 | ∞ | –9 | ∞ |
| 3 | –1 | 5 | 0 | **–4** | ∞ |
| 4 | ∞ | ∞ | ∞ | 0 | 8 |
| 5 | ∞ | ∞ | 9 | ∞ | 0 |

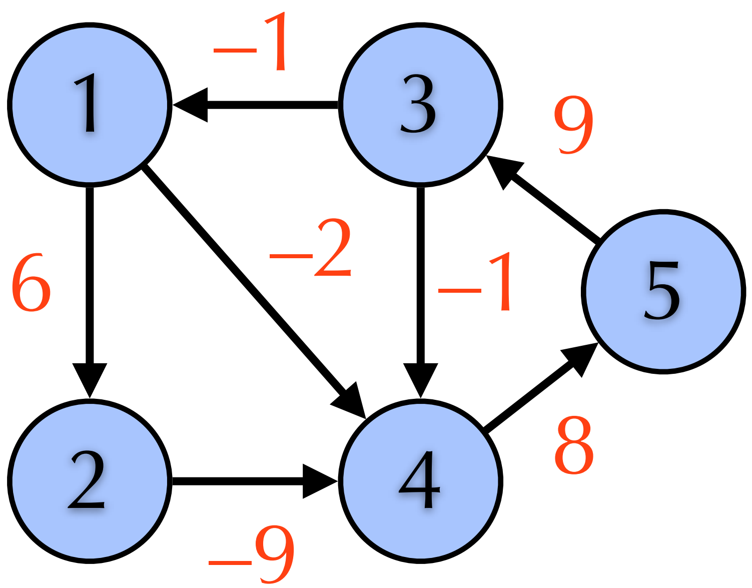
| $\Pi^2$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | NIL | 1 | NIL | **2** | NIL |
| 2 | NIL | NIL | NIL | 2 | NIL |
| 3 | 3 | 1 | NIL | **2** | NIL |
| 4 | NIL | NIL | NIL | NIL | 4 |
| 5 | NIL | NIL | 5 | NIL | NIL |




# Example

| $D^3$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 6 | ∞ | –3 | ∞ |
| 2 | ∞ | 0 | ∞ | –9 | ∞ |
| 3 | –1 | 5 | 0 | –4 | ∞ |
| 4 | ∞ | ∞ | ∞ | 0 | 8 |
| 5 | **8** | **14** | 9 | **5** | 0 |

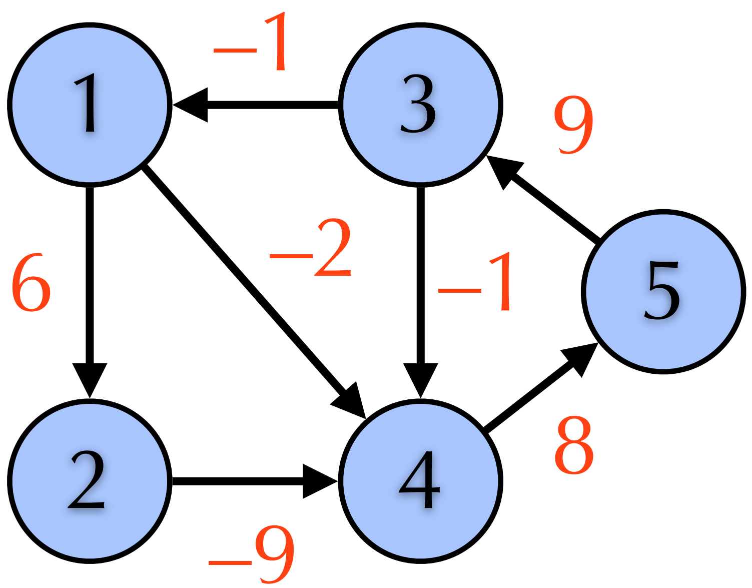
| $\Pi^3$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | NIL | 1 | NIL | 2 | NIL |
| 2 | NIL | NIL | NIL | 2 | NIL |
| 3 | 3 | 1 | NIL | 2 | NIL |
| 4 | NIL | NIL | NIL | NIL | 4 |
| 5 | **3** | **1** | 5 | **2** | NIL |




# Example

| $D^4$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 6 | ∞ | –3 | **5** |
| 2 | ∞ | 0 | ∞ | –9 | **–1** |
| 3 | –1 | 5 | 0 | –4 | **4** |
| 4 | ∞ | ∞ | ∞ | 0 | 8 |
| 5 | 8 | 14 | 9 | 5 | 0 |

| $\Pi^4$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | NIL | 1 | NIL | 2 | **4** |
| 2 | NIL | NIL | NIL | 2 | **4** |
| 3 | 3 | 1 | NIL | 2 | **4** |
| 4 | NIL | NIL | NIL | NIL | 4 |
| 5 | 3 | 1 | 5 | 2 | NIL |




# Example

| $D^5$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 6 | **14** | –3 | 5 |
| 2 | **7** | 0 | **8** | –9 | –1 |
| 3 | –1 | 5 | 0 | –4 | 4 |
| 4 | **16** | **22** | **17** | 0 | 8 |
| 5 | 8 | 14 | 9 | 5 | 0 |

| $\Pi^5$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | NIL | 1 | **5** | 2 | 4 |
| 2 | **3** | NIL | **5** | 2 | 4 |
| 3 | 3 | 1 | NIL | 2 | 4 |
| 4 | **3** | **1** | **5** | NIL | 4 |
| 5 | 3 | 1 | 5 | 2 | NIL |

# Complexity

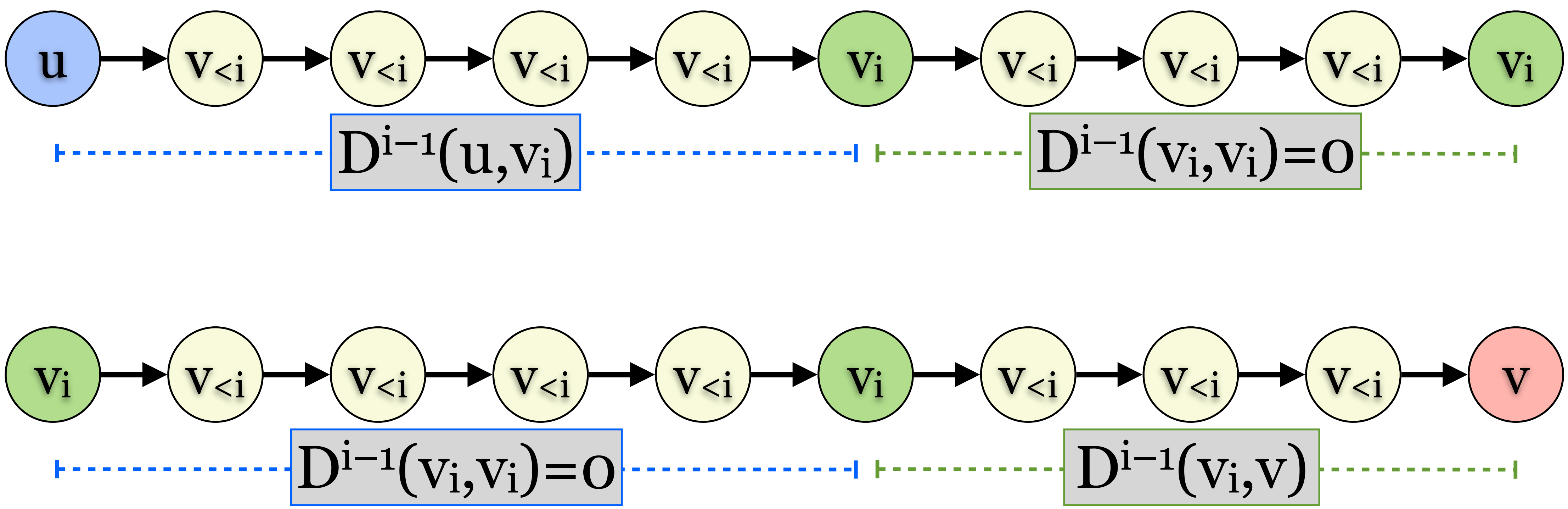
- Time: $\Theta(|V|^3)$
  - $\Theta(|V|^3)$ subproblems
  - $\Theta(1)$-time for each subproblem
- Space: $\Theta(|V|^3)$
  - $D^i$ takes $\Theta(|V|^2)$
    - $D^0,\ldots,D^n$ take $\Theta(|V|^3)$
  - Can be reduce to $\Theta(|V|^2)$
    - Use only $D$ and $\Pi$.

# Improvement: Space Complexity

- $D=W$
  $\Pi=\Pi^0$
  for i = 1 to n
      for $u \in V$
          for $v \in V$
              if $D(u,v)>D(u,v_i)+D(v_i,v)$
                  $D(u,v)=D(u,v_i)+D(v_i,v)$
                  $\Pi(u,v)=\Pi(v_i,v)$
  return D

# The Difference

- The new one might use $D^i(u,v_i)/D^i(v_i,v)$ instead of $D^{i-1}(u,v_i)/D^{i-1}(v_i,v)$.
- But $D^{i-1}(u,v_i)=D^i(u,v_i)$, $D^{i-1}(v_i,v)=D^i(v_i,v)$.



# Johnson's Algorithm

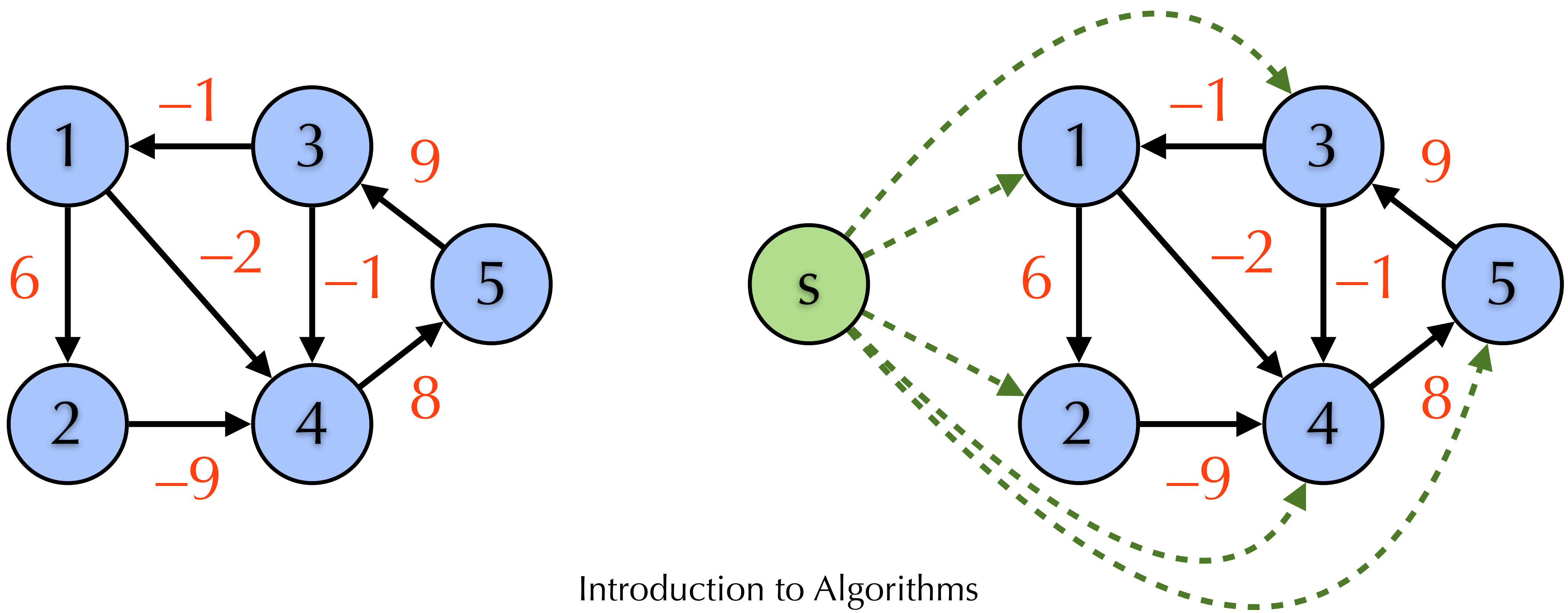
- Floyd-Warshall is simple and efficient if the graph is dense.
- Dijkstra's has better performance if the graph is sparse.
  - But Dijkstra's cannot handle negative edges.
- Johnson's:
  - Reweighting the graph by Bellman-Ford
    - No negative edges Triangle inequality
  - Apply Dijkstra's.

# Reweighting

- Idea: give a height $h(v)$ to vertex $v \in V$
  - Use Bellman-Ford
- New weight: $w'(u,v)=w(u,v)+h(u)-h(v)$
  - $w'(p)=w(p)+h(u)-h(v)$ if $p=\langle u=v_0,\ldots,v_k=v\rangle$ is a path from u to v.
    - $\Sigma_{1\le i\le k} w'(v_{i-1},v_i)= \Sigma_{1\le i\le k} w(v_{i-1},v_i)+h(v_{i-1})-h(v_i)$
- Shortest paths are not changed by reweighting.

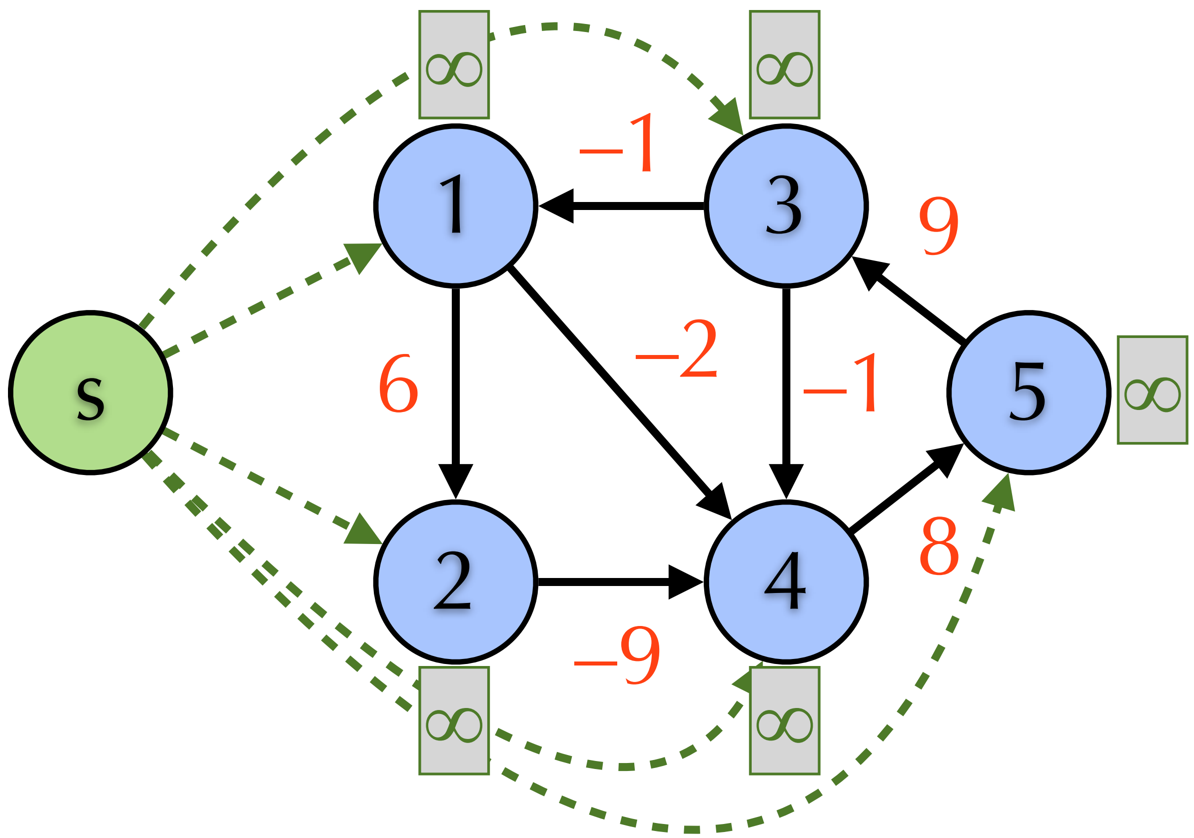
# Reweighting

- Goal: w'(u,v)≥0 for every (u,v)∈E
- Goal: w(u,v)≥h(v)−h(u)
- Add a vertex s and an edge (s,v) for v∈V.
  - w(s,v)=0



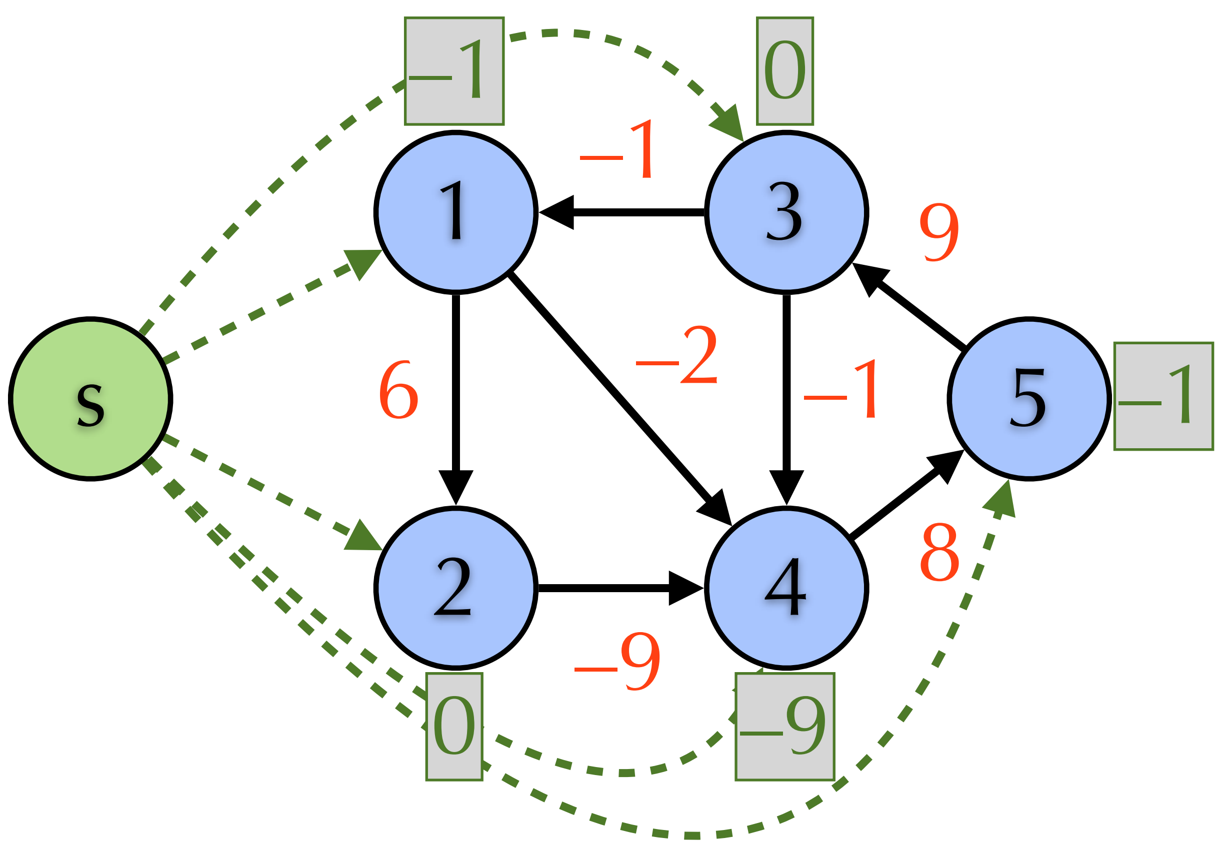
# Reweighting

- Run Bellman-Ford: source s



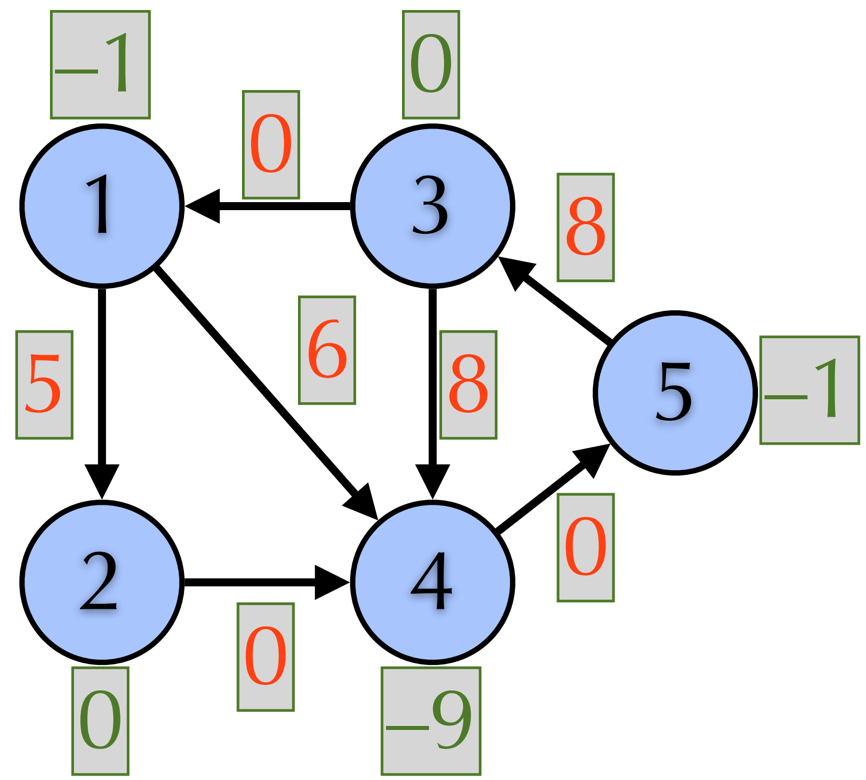
# Reweighting

- Run Bellman-Ford: source s DONE
- Set h(v)=δ(s,v)



# Reweighting

- Set w'(u,v)=w(u,v)+h(u)–h(v)≥0
  - w(u,v)+δ(s,u)≥δ(s,v)     Triangle inequality



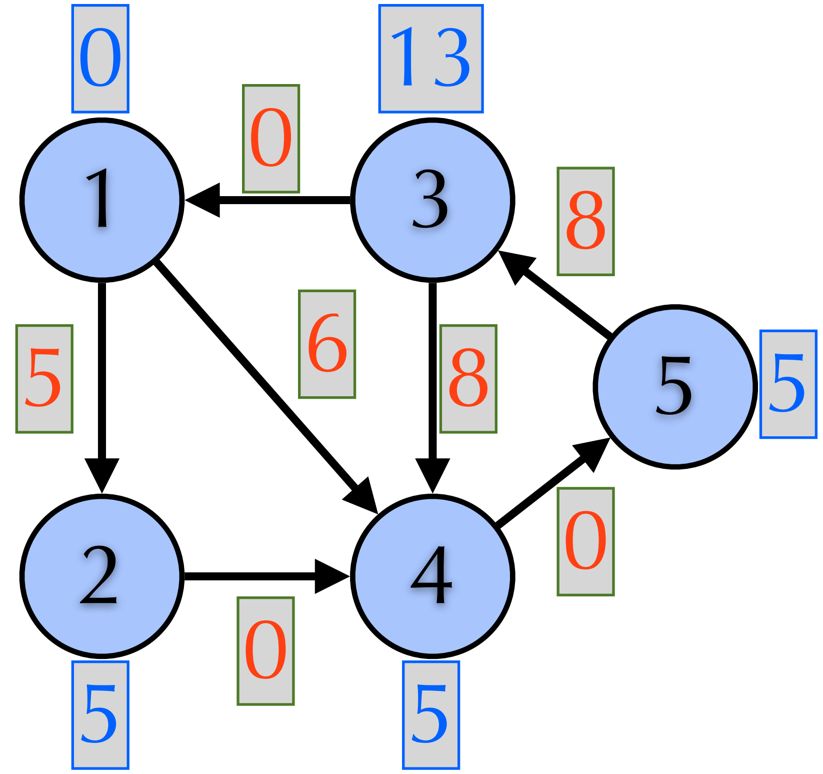

| v | h(v) |
|---|---|
| 1 | –1 |
| 2 | 0 |
| 3 | 0 |
| 4 | –9 |
| 5 | –1 |

$w'(p)=w(p)+h(u)-h(v)$

# Johnson's

- Run Dijkstra's: source 1

| δ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 6 | 14 | –3 | 5 |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |



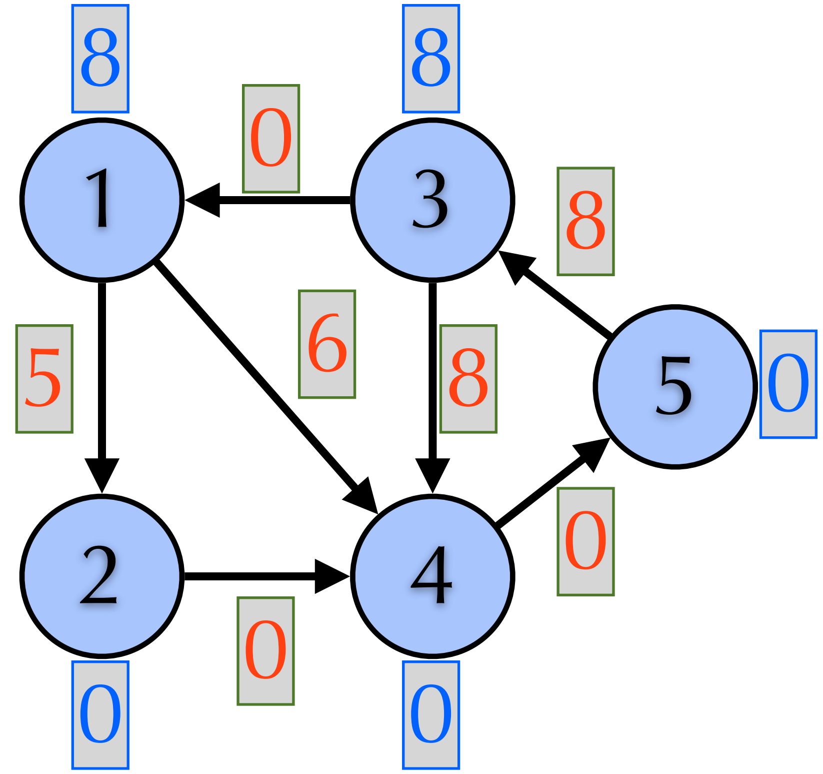

| v | h(v) |
|---|---|
| 1 | –1 |
| 2 | 0 |
| 3 | 0 |
| 4 | –9 |
| 5 | –1 |

$w'(p)=w(p)+h(u)-h(v)$

# Johnson's

- Run Dijkstra's: source 2

| δ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 6 | 14 | –3 | 5 |
| 2 | 7 | 0 | 8 | –9 | –1 |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |



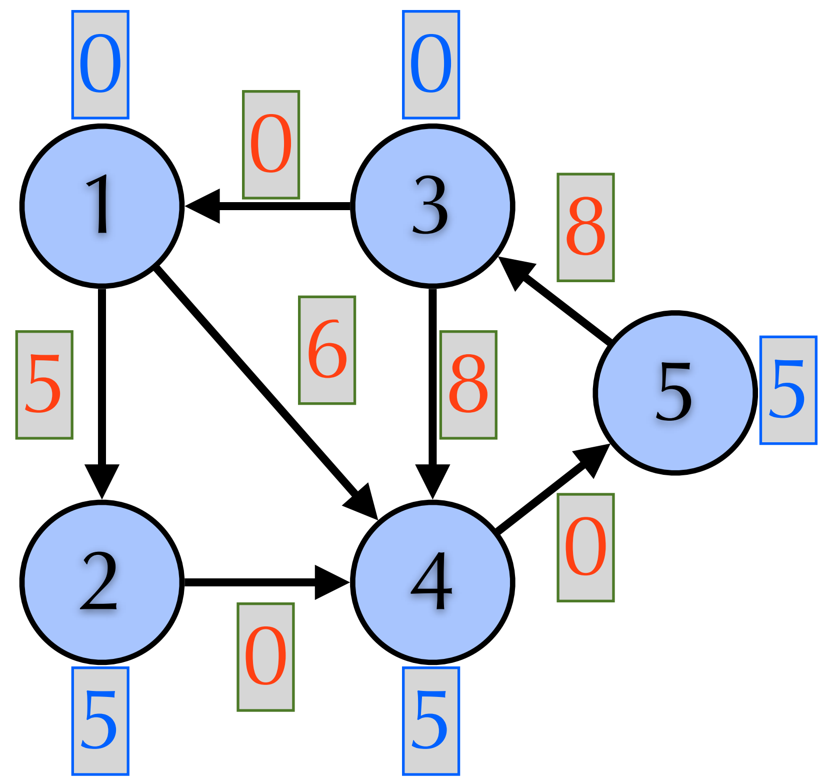

| v | h(v) |
|---|---|
| 1 | –1 |
| 2 | 0 |
| 3 | 0 |
| 4 | –9 |
| 5 | –1 |

w’(p)=w(p)+h(u)–h(v)

# Johnson’s

- Run Dijkstra’s: source 3

| δ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 6 | 14 | –3 | 5 |
| 2 | 7 | 0 | 8 | –9 | –1 |
| 3 | –1 | 5 | 0 | –4 | 4 |
| 4 | | | | | |
| 5 | | | | | |



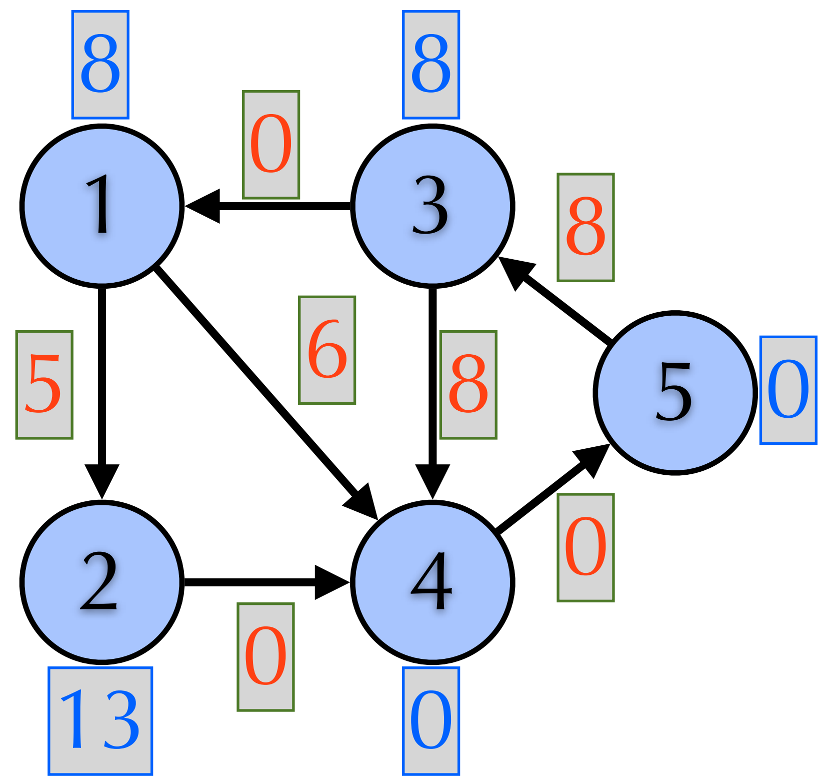

| v | h(v) |
|---|---|
| 1 | –1 |
| 2 | 0 |
| 3 | 0 |
| 4 | –9 |
| 5 | –1 |

$w'(p)=w(p)+h(u)-h(v)$

# Johnson's

- Run Dijkstra's: source 4

| δ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 6 | 14 | –3 | 5 |
| 2 | 7 | 0 | 8 | –9 | –1 |
| 3 | –1 | 5 | 0 | –4 | 4 |
| 4 | 16 | 22 | 17 | 0 | 8 |
| 5 | | | | | |



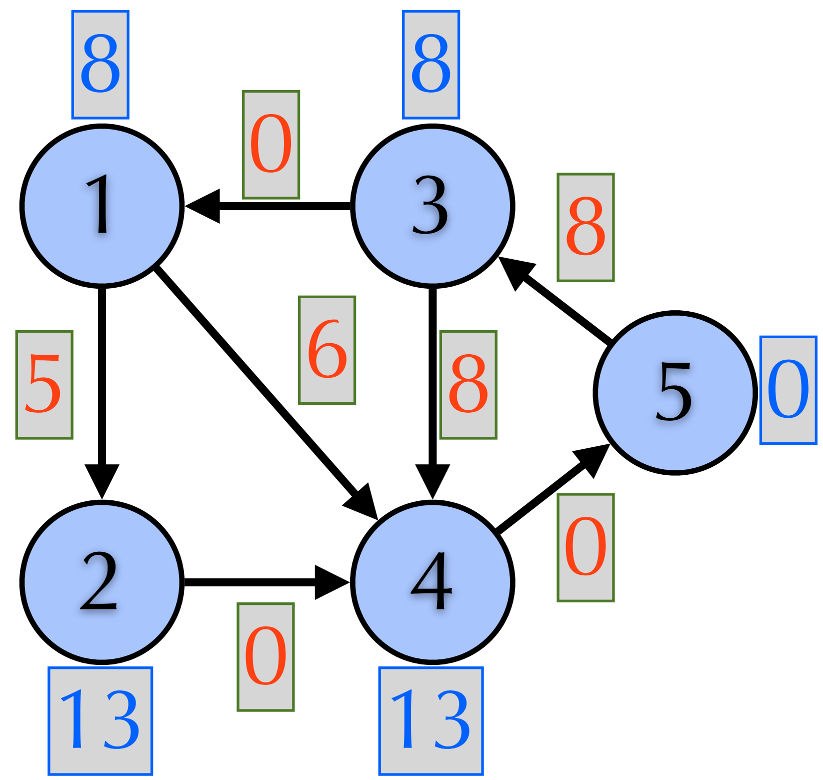

| v | h(v) |
|---|---|
| 1 | –1 |
| 2 | 0 |
| 3 | 0 |
| 4 | –9 |
| 5 | –1 |

$w'(p)=w(p)+h(u)-h(v)$

# Johnson's

- Run Dijkstra's: source 5

| δ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 6 | 14 | –3 | 5 |
| 2 | 7 | 0 | 8 | –9 | –1 |
| 3 | –1 | 5 | 0 | –4 | 4 |
| 4 | 16 | 22 | 17 | 0 | 8 |
| 5 | 8 | 14 | 9 | 5 | 0 |




| v | h(v) |
|---|---|
| 1 | –1 |
| 2 | 0 |
| 3 | 0 |
| 4 | –9 |
| 5 | –1 |

# Time Complexity

- Bellman-Ford: $O(|V||E|)$
- $|V| \times$ Dijkstra's:
  - $O(|V|^3)$ Array
  - $O(|V||E|\log|V|)$ Binary heap
  - $O(|V|^2\log|V|+|V||E|)$ Fibonacci heap