Dynamic Programming

Hengfeng Wei

hfwei@nju.edu.cn

June 11 – June 19, 2017

Dynamic Programming

3-D DP

Floyd-Warshall algorithm

(1) DP for Floyd-Warshall algorithm for APSP on directed graphs

Subproblem: dist$[i, j, k]$: the length of the shortest path from i to j via only nodes in $v_1 \cdots v_k$

Goal: dist$[i, j, n], \forall i, j$

3-D DP

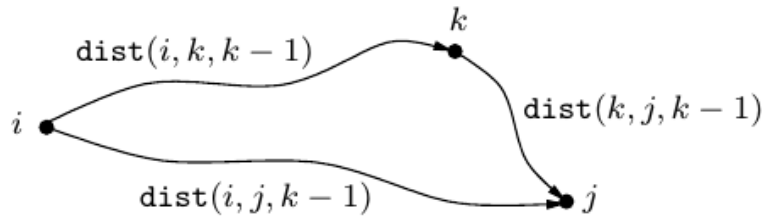
Floyd-Warshall algorithm

(1) DP for Floyd-Warshall algorithm for APSP on directed graphs

Make choice: Is v_k on the ShortestPath$[i, j, k]$?

Recurrence:

$$\text{dist}[i, j, k] = \min\{\text{dist}[i, j, k-1], \text{dist}[i, k, k-1] + \text{dist}[k, j, k-1]\}$$



3-D DP

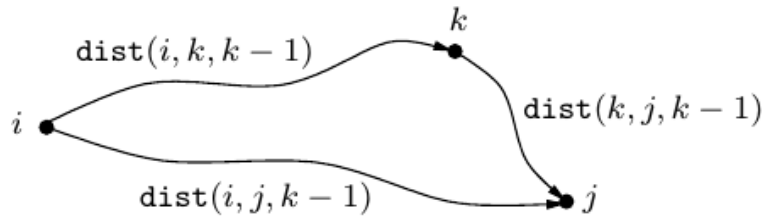
Floyd-Warshall algorithm

(1) DP for Floyd-Warshall algorithm for APSP on directed graphs

Make choice: Is v_k on the ShortestPath$[i, j, k]$?

Recurrence:

$$\text{dist}[i, j, k] = \min\{\text{dist}[i, j, k-1], \text{dist}[i, k, k-1] + \text{dist}[k, j, k-1]\}$$



Init:

$$\text{dist}[i, j, 0] = \begin{cases} 0 & i = j \\ w(i, j) & (i, j) \in E \\ \infty & \text{o.w.} \end{cases}$$

3-D DP

Floyd-Warshall algorithm (Problem 6.25)

(2) Routing table

for all $k \leftarrow 1 \ldots n$ **do**
  for all $i \leftarrow 1 \ldots n$ **do**
    for all $j \leftarrow 1 \ldots n$ **do**
      if $\text{dist}[i,j] > \text{dist}[i,k] + \text{dist}[k,j]$ **then**
        $\text{dist}[i,j] \leftarrow \text{dist}[i,k] + \text{dist}[k,j]$

3-D DP

Floyd-Warshall algorithm (Problem 6.25)

(2) Routing table

for all $k \leftarrow 1 \ldots n$ **do**
  for all $i \leftarrow 1 \ldots n$ **do**
    for all $j \leftarrow 1 \ldots n$ **do**
      if $\text{dist}[i,j] > \text{dist}[i,k] + \text{dist}[k,j]$ **then**
        $\text{dist}[i,j] \leftarrow \text{dist}[i,k] + \text{dist}[k,j]$
        $\text{Go}[i,j] \leftarrow \text{Go}[i,k]$

3-D DP

Floyd-Warshall algorithm (Problem 6.25)

(2) Routing table

for all $i \leftarrow 1 \ldots n$ **do**
    for all $j \leftarrow 1 \ldots n$ **do**
        $\text{dist}[i, j] \leftarrow \infty$
        $\text{Go}[i, j] \leftarrow \text{Nil}$
for all $(i, j) \in E$ **do**
    $\text{dist}[i, j] \leftarrow w(i, j)$
    $\text{Go}[i, j] \leftarrow j$
for all $i \leftarrow 1 \ldots n$ **do**
    $\text{dist}[i, i] \leftarrow 0$
    $\text{Go}[i, j] \leftarrow \text{Nil}$

3-D DP

Floyd-Warshall algorithm (Problem 6.25)

(2) Routing table

for all $i \leftarrow 1 \ldots n$ **do**
  for all $j \leftarrow 1 \ldots n$ **do**
    $\text{dist}[i, j] \leftarrow \infty$
    $\text{Go}[i, j] \leftarrow \text{Nil}$
for all $(i, j) \in E$ **do**
  $\text{dist}[i, j] \leftarrow w(i, j)$
  $\text{Go}[i, j] \leftarrow j$
for all $i \leftarrow 1 \ldots n$ **do**
  $\text{dist}[i, i] \leftarrow 0$
  $\text{Go}[i, j] \leftarrow \text{Nil}$

procedure $\textsc{Path}(i, j)$
  if $\text{Go}[i, j] = \text{Nil}$ **then**
    Output "No Path."

Output "i"
while $i \neq j$ **do**
  $i \leftarrow \text{Go}[i, j]$
Output "i"

3-D DP

Floyd-Warshall algorithm (Problem 6.29)

(3) Find Minimum-weighted cycle

$$\text{dist}[i,i] \leftarrow 0 \implies \text{dist}[i,i] \leftarrow \infty$$

`https://cs.stackexchange.com/q/76578/4911`

Dynamic Programming

Dynamic Programming


THANK YOU