Saving Space in the LCS Algorithm

COMP 3711H - HKUST
Version of 9/11/2016
M. J. Golin

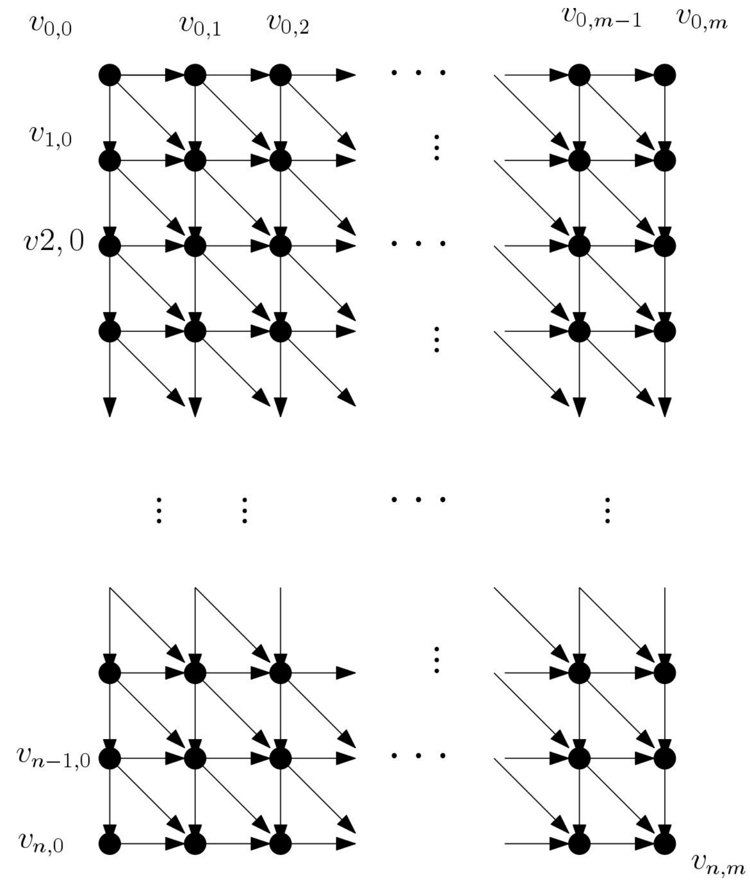
Problem is to calcuate value of $d_{n,m}$ defined by

$$d_{i,j} = \max\left(d_{i-1,j},\, d_{i,j-1},\, d_{i,j} + \delta_{i,j}\right)$$

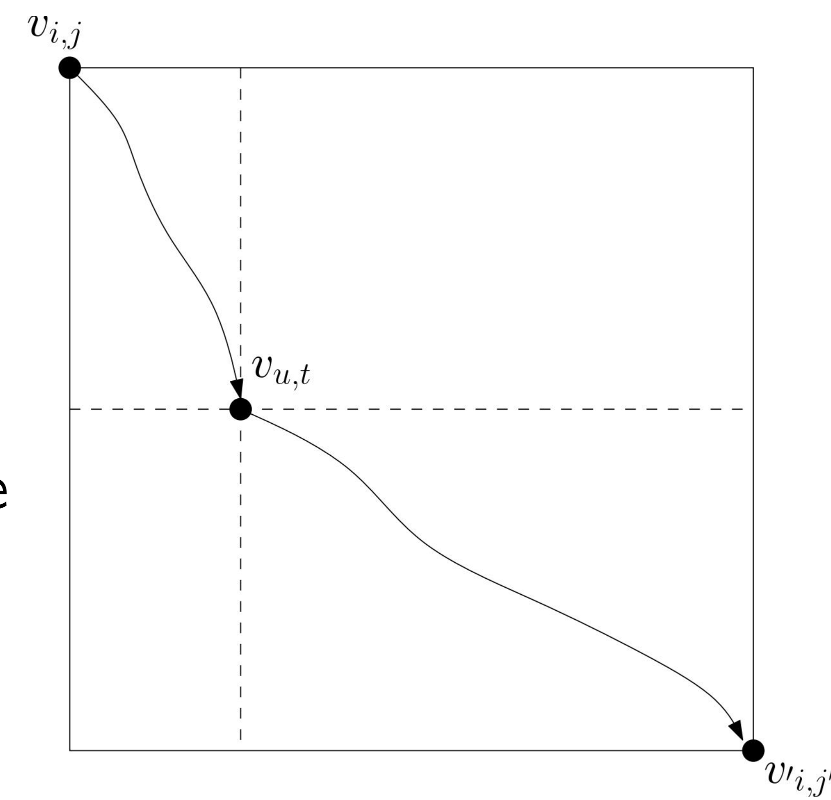
where

$$\delta_{i,j} = \begin{cases} 1 & \text{if } x[i] = y[j] \\ 0 & \text{if } x[i] \neq y[j]. \end{cases}$$

whic is same as finding max-cost path from $v_{0,0}$ to $v_{n.m}$ in grid graph with left and down edges having cost 0 and diagonal down edges from $v_{i,j}$ having cost $\delta_{i,j}$.



1. $BP(i, j, i, j')$
2. if $j' = j$ then
3. return $BP(i, j, i', j)$ by writing the vertical path in $O(i' - i + 1)$ time
4. if $i' = i$ then
5. return $BP(i, j, i, j')$ by writing the horizontal path in $O(j' - j + 1)$ time
6. else if $i' = i + 1$
7. return $BP(i, j, i, j')$ by running the DAG algorithm in $O(j' - j + 1)$ time
8. else
9. Run $Mid(i, j, i + 1, j')$ to find (u, v) with $u = \lfloor \frac{i+i'}{2} \rfloor$
10. return $BP(i, j, u, v)$ $BP(u, v, i', j')$



$Mid(i, j, i', j')$ returns a vertex (u, v) on some max cost path between $v_{i,j}$ and $v_{i',j'}$ with $u = \lfloor \frac{i+i'}{2} \rfloor$.

$BP(i, j, i, j')$
if $j' = j$ then
    return $BP(i, j, i', j)$ by writing the verticle path in $O(i' - i + 1)$ time
if $i' = i$ then
    return $BP(i, j, i, j')$ by writing the horizontal path in $O(j' - j + 1)$ time
else if $i' = i + 1$
    return $BP(i, j, i, j')$ by running the DAG algorithm in $O(j' - j + 1)$ time
else
    Run $Mid(i, j, i + 1, j')$ to find (u, v) with $u = \lfloor \frac{i+i'}{2} \rfloor$
    return $BP(i, j, u, v)$ $BP(u, v, i', j')$

If $Mid(i, j, i', j')$ uses $Perim(Box(i, j, i', j')$ space then $BP(i, j, i, j')$ also uses $Perim(Box(i, j, i', j')$ space.

If $Mid(i, j, i', j')$ uses $Area(Box(i, j, i', j')$ time then $BP(i, j, i, j')$ also uses $Area(Box(i, j, i', j')$ time

$Mid(i,j,i',j')$ returns a vertex (u,t) on some max cost path between $v_{i,j}$ and $v_{i',j'}$ with $u = \lfloor \frac{i+i'}{2} \rfloor$.

Let G' be the induced subgraph of G containing all of the

$$\{v_{s,t} \,|\, i \le s \le v',\ j \le t \le j'\}$$

Let G_1 be induced subgraph of G' containing all $v_{s,t}$ with $s \le u$.
Let $d^1_{s,t}$ be the cost of a max cost path from $v_{i,j}$ to $v_{s,t}$ in G_1.
Using the same approach used for the LCS algorithm we can walk down row by row calculating $d^1_{s,*}$ (i.e., the entire s row's values) from $d^1_{s-1,*}$. This uses $O(Area(Box(i,j,i',j'))$ time and $O(Perim(Box(i,j,i',j'))$ space.

When finished, we have stored (only) all of the values $d^1_{u,*}$

$Mid(i, j, i', j')$ returns a vertex (u, t) on some max cost path between $v_{i,j}$ and $v_{i',j'}$ with $u = \lfloor \frac{i+i'}{2} \rfloor$.

Let G' be the induced subgraph of G containing all of the

$$\{v_{s,t} \,|\, i \le s \le v',\ j \le t \le j'\}$$

Let G'_2 be induced subgraph of G' containing all $v_{s,t}$ with $s \le u$ and *all edges reversed*.

Let $d^2_{s,t}$ be the cost of a max cost path from $v_{n,m}$ to $v_{s,t}$ in G'_2.

Using the same approach used for the LCS algorithm we can walk up row by row calculating $d^2_{s,*}$ (i.e., the entire s row's values) from $d^1_{s+1,*}$. This uses $O(Area(Box(i, j, i', j'))$ time and $O(Perim(Box(i, j, i', j'))$ space.

When finished, we have stored (only) all of the values $d^2_{u,*}$

For every $j \le t \le j'$, $d^1_{u,t} + d^2_{u,t}$ is the cost of max cost path from $v_{i,j}$ to $v_{i',j'}$ *passing through* $v_{u,t}$. Since a max cost path from $v_{i,j}$ to $v_{i',j'}$ must pass through at least one node $v_{u,t}$ the value

$$\max_t \left(d^1_{u,t} + d^2_{u,t}\right)$$

is the cost of a max-cost path from from $v_{i,j}$ to $v_{i',j'}$. If t' is the index t at which the maximum occurs then all our procedure needs to do is return the node (u, t').

This last step only used $O\left(j' - j + 1\right)$ time and $O\left(j' - j + 1\right)$ space so the entire procedure for $Mid(i, j, i', j')$ only used $O(Area(Box(i, j, i', j'))$ time and $O(Perim(Box(i, j, i', j'))$ space and we are done.