ALL PAIRS SHORTEST PATHS

using

RECTANGULAR MATRIX MULTIPLICATIONS

and

BRIDGING SETS

Uri Zwick

Tel Aviv University

THE PROBLEM

$G = (V, E)$ —

A weighted directed graph on n vertices with weights taken from the set $\{-M, \ldots -1, 0, 1, \ldots, M\}$.

Find

SHORTEST PATHS

between all pairs of vertices in the graph.

Directed graphs with real weights

Dijkstra '59 Johnson '77 Fredman, Tarjan '87	$nm + n^2 \log n$
Karger, Koller, Phillips '93 McGeoch '95	$nm^* + n^2 \log n$
Fredman '76 Takaoka '92	$n^3 \left(\frac{\log\log n}{\log n}\right)^{1/2}$

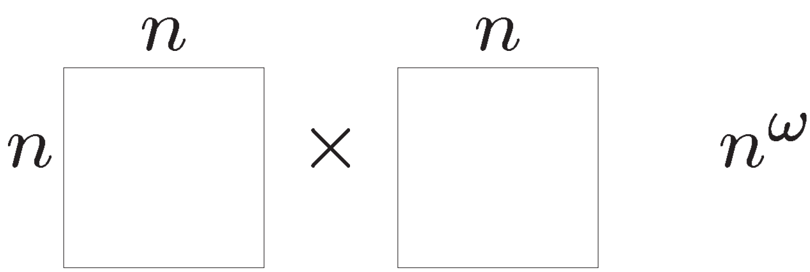
Directed graphs with small integer weights

Alon, Galil Margalit '91	$n^{\frac{3+\omega}{2}} \le n^{2.688}$
Zwick '98	$n^{2+\frac{1-\alpha(\omega-1)}{4-\omega-2\alpha}} \le n^{2.575}$

Undirected graphs with small integer weights

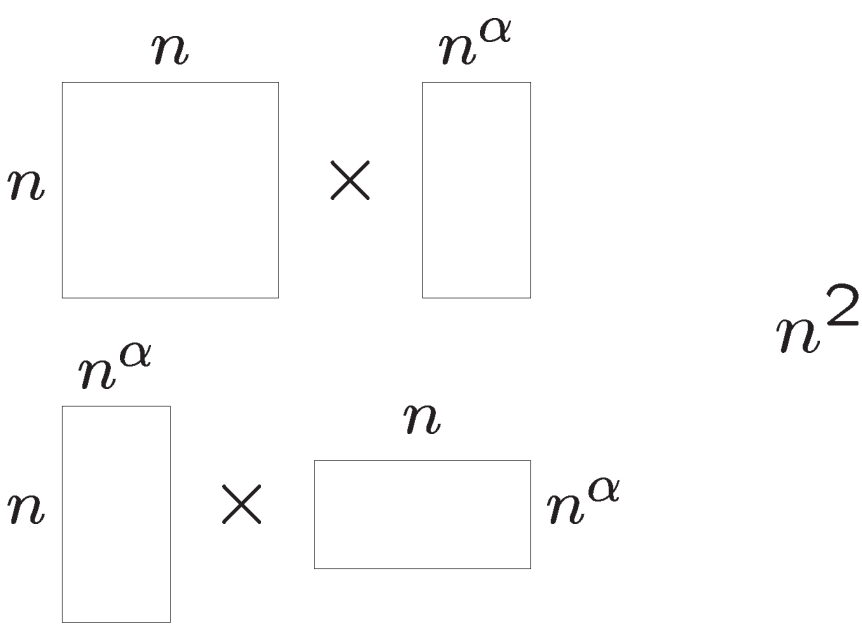
Seidel '92	$n^\omega \le n^{2.376}$
Galil, Margalit '92	$n^\omega \le n^{2.376}$

SQUARE MATRIX MULTIPLICATION



$\omega < 2.376$ [Coppersmith, Winograd '90]

RECTANGULAR MATRIX MULTIPLICATION



$\alpha > 0.294$ [Coppersmith '97]

OVER ANY RING

$$\frac{3+\omega}{2} \leq 2.688$$

$$2+\frac{1-\alpha(\omega-1)}{4-\omega-2\alpha} \leq 2.575$$

$$2+\frac{1}{4-\omega} \leq 2.616$$

If $\omega = 2$ then the running time of both algorithms becomes $n^{2.5}$

To achieve a running time of $n^{2.5}$ with the new algorithm it is enough to show that $\alpha \geq 1/2$.

APPROXIMATION ALGORITHMS

Unweighted, undirected graphs

Aingworth, Chekuri Indyk, Motwani	'96	$n^{5/2}$	$+2$
Dor, Halperin Zwick	'96	$n^{7/3}$	$+2$

Weighted, undirected graphs

Cohen Zwick	'97	$n^{3/2}m^{1/2}$	$\times 2$
		n^2	$\times 3$

Weighted, directed graphs

Zwick '98 $\quad \dfrac{n^\omega}{\epsilon} \cdot \log\left(\dfrac{W}{\epsilon}\right) \quad \times (1+\epsilon)$

DISTANCE PRODUCTS

$$\begin{bmatrix} -6 & -3 & -10 \\ 2 & 5 & -2 \\ -1 & -7 & -5 \end{bmatrix} = \begin{bmatrix} 1 & -3 & 7 \\ +\infty & 5 & +\infty \\ 8 & 2 & -5 \end{bmatrix} \star \begin{bmatrix} 8 & +\infty & -4 \\ -3 & 0 & -7 \\ 5 & -2 & 1 \end{bmatrix}$$

$$C \qquad = \qquad A \qquad \star \qquad B$$

$$c_{ij} = \min_k \{ a_{ik} + b_{kj} \}$$

If D contains the edge weights of a graph

$$D = \begin{bmatrix} 0 & & & 5 & -8 \\ & 0 & & & +\infty \\ & & \ddots & & \\ +\infty & & & 0 & \\ 7 & -5 & & & 0 \end{bmatrix}$$

Then D^n contains all the distances.

CAN WE USE

FAST MATRIX MULTIPLICATION

ALGORITHMS

DIRECTLY

TO COMPUTE

DISTANCE PRODUCTS ?

NO !

$(\mathbb{R}, \min, +)$

IS NOT A RING

min has no inverse

Using matrix multiplication to compute distance products (Yuval '76)

$$\begin{bmatrix} c_{11} & c_{12} & \\ c_{21} & c_{22} & \\ & & \ddots \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \\ a_{21} & a_{22} & \\ & & \ddots \end{bmatrix} \star \begin{bmatrix} b_{11} & b_{12} & \\ b_{21} & b_{22} & \\ & & \ddots \end{bmatrix}$$

$$c_{ij} = \max_k \{ \, a_{ik} + b_{kj} \, \}$$

$$0 \le a_{ij}, b_{ij} \le M$$

$$C' = \begin{bmatrix} x^{a_{11}} & x^{a_{12}} & \\ x^{a_{21}} & x^{a_{22}} & \\ & & \ddots \end{bmatrix} \star \begin{bmatrix} x^{b_{11}} & x^{b_{12}} & \\ x^{b_{21}} & x^{b_{22}} & \\ & & \ddots \end{bmatrix}$$

x - an indeterminant

Compute the normal algebraic product of the two polynomial matrices

$$c'_{ij} = \sum_k x^{a_{ik}+b_{kj}}$$

$$c_{ij} = \deg(c'_{ij})$$

$$n^\omega$$

polynomial products

$$\times$$

$$M \cdot \log n$$

time per each
polynomial product

$$\Downarrow$$

The distance products of
two $n \times n$ matrices
with elements taken from
$\{-M, \ldots, 0, \ldots, M\} \cup \{+\infty\}$
can be computed in about

$$M \cdot n^\omega$$

time

WITNESSED DISTANCE PRODUCTS

$$C = A \star B$$

$$c_{ij} = \min_k \{ \, a_{ik} + b_{kj} \, \}$$

$$c_{ij} = a_{i,w_{ij}} + b_{w_{ij},j}$$

$(C, W) \leftarrow$ **DIST-PROD**(A, B, M)

Replace all entries in A and B of absolute value $>$ M by $+\infty$ and compute the product and a matrix of witnesses.

REPEATED POWERING

Suppose all weights are 1.

All entries in D are from

$\{1, +\infty\}$

$F \leftarrow D$
repeat $\lceil \log_2 n \rceil$ **times**
$F \leftarrow F \star F$

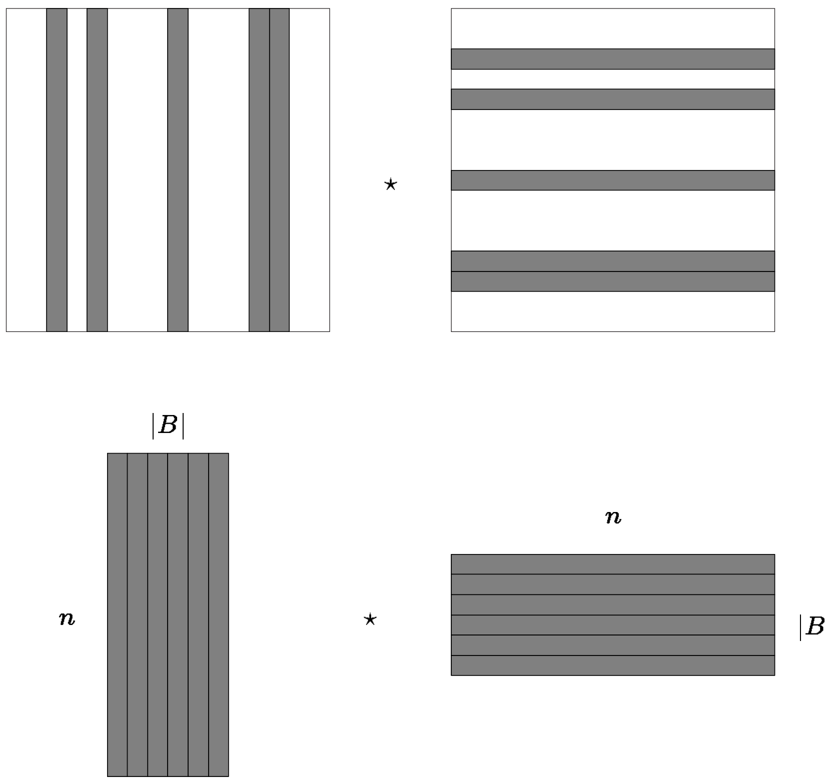
Naive implementation takes

$n^3 \log n$ time

Using the $M \cdot n^\omega$ algorithm

we get $n \cdot n^\omega = n^{\omega+1} \gg n^3$

REPEATED POWERING with SAMPLING



In each iteration, Choose a subset B of the vertices and replace the product

$$F \star F$$

by

$$F[*, B] \star F[B, *]$$

A RANDOMIZED ALGORITHM

```
algorithm RAND-SHORT-PATH(D)
```

$F \leftarrow D$; $W \leftarrow 0$

$M \leftarrow \max\{\, |d_{ij}| : d_{ij} \neq +\infty\}$

for $\ell \leftarrow 1$ to $\lceil \log_{3/2} n \rceil$ do

begin

  $s \leftarrow (3/2)^{\ell}$

  $B \leftarrow$ **RAND**$(\{1, 2, \ldots, n\}, (9 \ln n)/s)$

  $(F', W') \leftarrow$ **DIST-PROD**$(F[*, B], F[B, *], sM)$

  for every $1 \leq i, j \leq n$ do

  if $f'_{ij} < f_{ij}$ then $f_{ij} \leftarrow f'_{ij}$; $w_{ij} \leftarrow b_{w'_{ij}}$

end

return F

CORRECTNESS

At any stage,

for any i and j,

$$f_{ij} \geq \delta(i,j)$$

After the ℓ-th iteration,

with **HIGH PROBABILITY**,

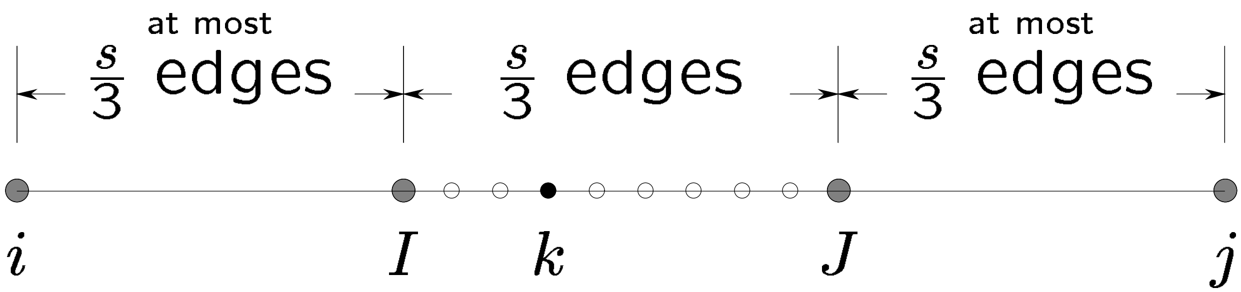
if there is a **SHORTEST PATH** from i to j that uses at most $s = (3/2)^\ell$ edges, then

$$f_{ij} = \delta(i,j)$$

PROOF OF CORRECTNESS

Suppose the claim holds
after the $(\ell - 1)$-st iteration.

Suppose there is a shortest path from i to j
that uses at most s edges.



With HIGH PROBABILITY
$\exists\ k \in B$ between I and J.

$$\left(1 - \frac{9 \ln n}{s}\right)^{s/3} \leq \mathbf{e}^{-3 \ln n} = n^{-3}$$

Thus,

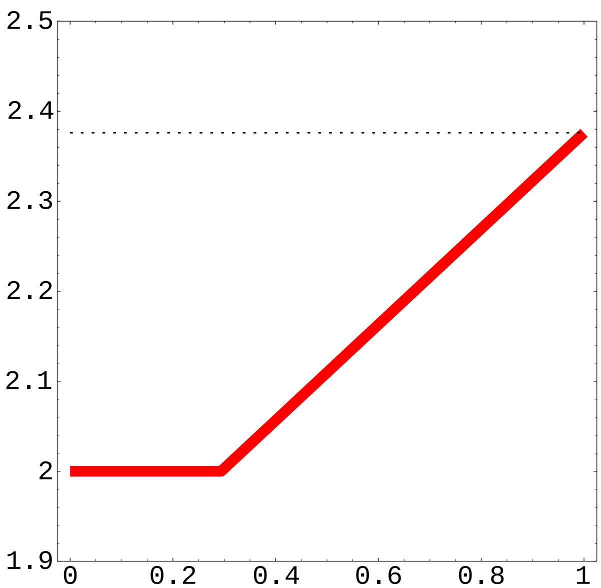
$$\begin{aligned} f_{ij} &\leq f_{ik} + f_{kj} \\ &= \delta(i,k) + \delta(k,j) \\ &= \delta(i,j) \end{aligned}$$

COMPLEXITY ANALYSIS

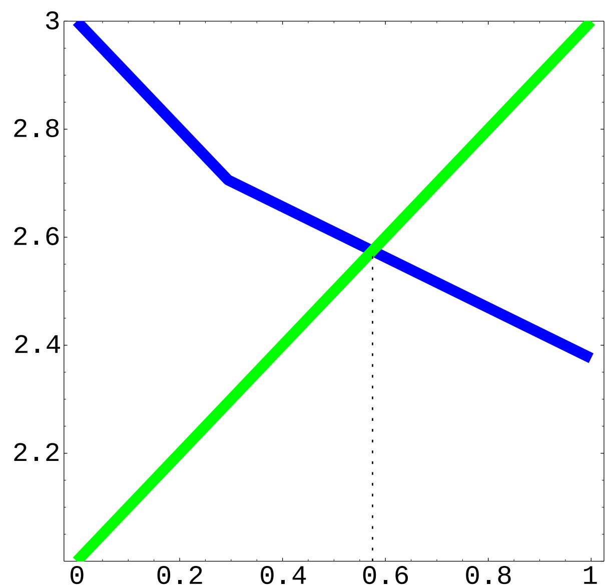
$$s = (3/2)^{\ell} = n^{1-r}$$

$n \times n^r$ by $n^r \times n$
product

$$\min\{\ n^{1-r} \cdot M \cdot n^{\omega(1,r,1)}\ ,\ n^{2+r}\ \}$$



$\omega(1,r,1)$



$(1-r)+\omega(1,r,1)$
vs.
$2+r$

$$(1-r)+\omega(1,r,1) \ = \ 2+r$$

$$\omega(1,r,1) \ = \ 1+2r$$

$r \approx 0.575$

FINDING SHORTEST PATHS

```
algorithm PATH(W, i, j)

if w_ij = j then
   return ⟨i, j⟩
else
   return PATH(W, i, w_ij) . PATH(W, w_ij, j)
endif
```

DETERMINISTIC

ALGORITHM

?

BRIDGING SETS

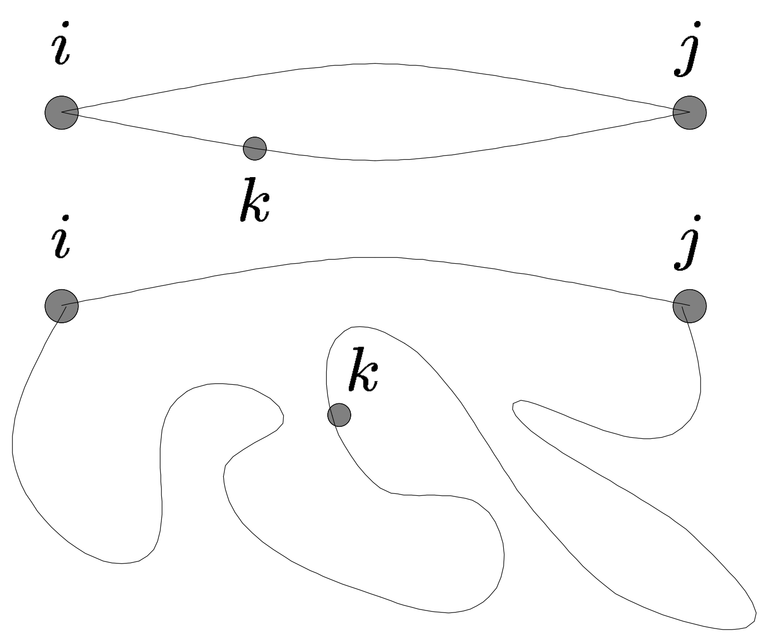
$\delta(i,j)$ – the distance from i to j
$\eta(i,j)$ – the minimal number of edges on a shortest path from i to j

A set of vertices B is

[STRONGLY] s-BRIDGING

iff

$$\forall\, i,j \in V \qquad \eta(i,j) \geq s \Longrightarrow$$
$$\exists\, k \in B\,, \quad \delta(i,j) = \delta(i,k) + \delta(k,j)$$
$$[\,\eta(i,j) = \eta(i,k) + \eta(k,j)\,]$$



A DETERMINISTIC ALGORITHM FOR FINDING BRIDGING SETS

```
algorithm FIND-BRIDGE(W, s)
C ← φ
for every 1 ≤ i, j ≤ n do
      U ← SUB-PATH(W, i, j, s)
      if |U| ≥ s + 1 then C ← C ∪ {U}
end
B ← HITTING-SET(C)
return B
```

Algorithm **FIND-BRIDGE** runs in $O(n^2 s)$ time and finds an s-bridging set of size at most $(2n \ln n)/s$.

DEPENDENCY OF RUNNING TIME ON M

$$n^{2+\frac{1-\alpha(\omega-1)}{4-\omega-2\alpha}} \cdot M^{\frac{1-\alpha}{4-\omega-2\alpha}}$$

$$n^{2.575} \cdot M^{0.681}$$

SUB-CUBIC for

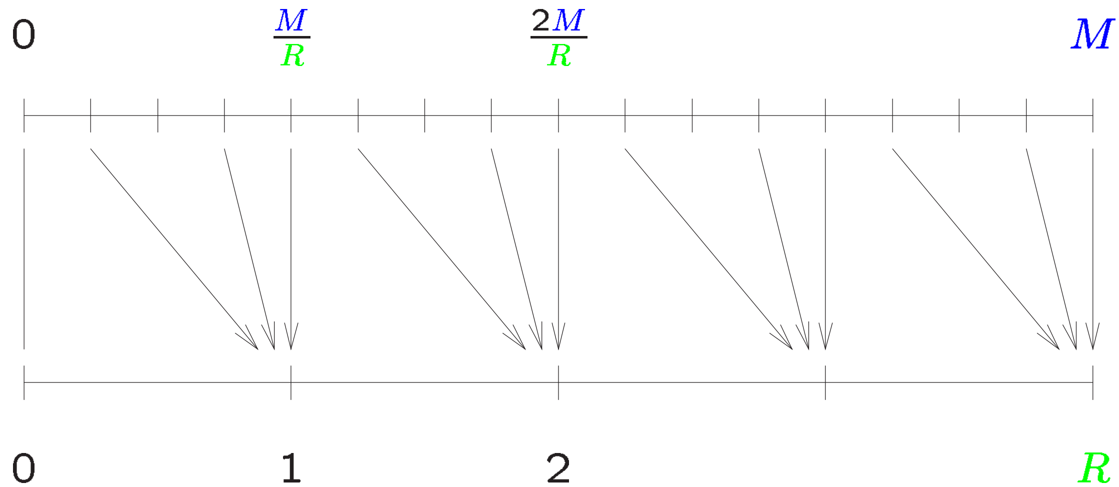
$$M < n^{3-\omega}$$

$$M < n^{0.624}$$

SCALING

SCALE(A, M, R) :

$$a'_{ij} \leftarrow \begin{cases} \lceil R\, a_{ij}/M \rceil & \text{if } \ 0 \le a_{ij} \le M \\ +\infty & \text{othersize} \end{cases}$$

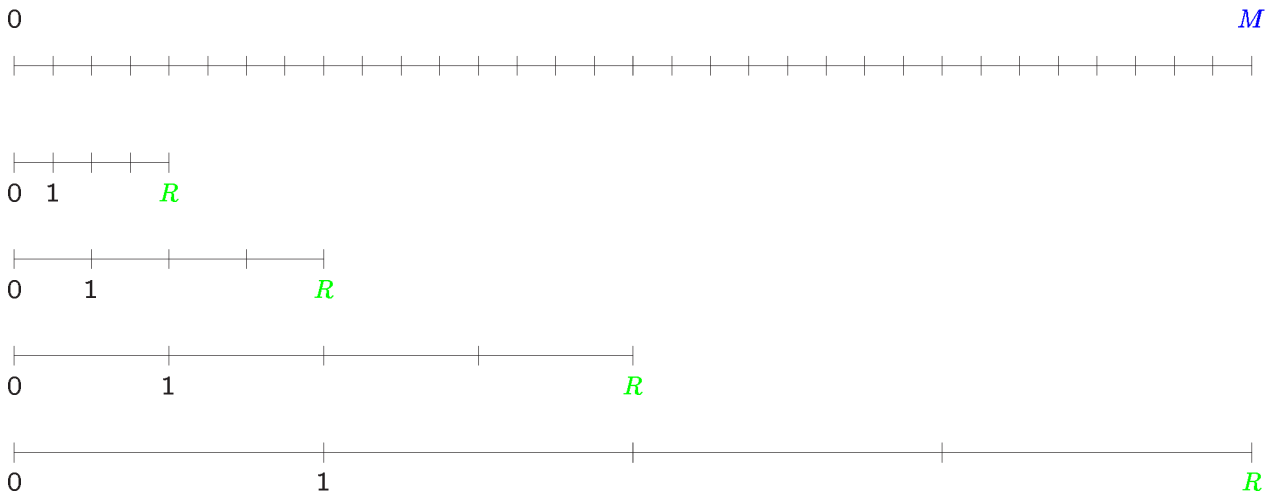


APX-DIST-PROD(A, M, R) :

$A' \leftarrow$ **SCALE**(A, M, R)
$B' \leftarrow$ **SCALE**(B, M, R)
$C' \leftarrow$ **DIST-PROD**(A', B', R)
$C \leftarrow (M/R) \cdot C'$
`return` C

We may get a HUGE stretch of $\frac{M}{R}$

ADAPTIVE SCALING



algorithm **APX-DIST-PROD**(A, B, M, R)

$C \leftarrow +\infty$

for $r \leftarrow \log_2 R$ **to** $\log_2 M$ **do**

begin

$\quad A' \leftarrow \textbf{SCALE}(A, 2^r, R)$

$\quad B' \leftarrow \textbf{SCALE}(B, 2^r, R)$

$\quad C' \leftarrow \textbf{DIST-PROD}(A', B', R)$

$\quad C \leftarrow \min\{\, C \,,\ (2^r/R) \cdot C' \,\}$

end

return C

Stretch at most $1 + \frac{4}{R}$

APPROXIMATE SHORTEST PATHS

```
algorithm APPROX-SHORT-PATH(D, ε)

F ← D
M ← max{ d_ij : d_ij ≠ +∞}
R ← (4 log_2 n)/ ln(1 + ε) ; R ← 2^⌈log_2 R⌉

for ℓ ← 1 to ⌈log_2 n⌉ do
begin
    F' ← APX-DIST-PROD(F, F, Mn, R)
    F ← min{ F , F' }
end

return F
```

After $\log n$ iterations, the stretch is at most

$$\left(1+\frac{4}{R}\right)^{\log n} \leq 1+\epsilon$$

COMPLEXITY:

$$n^{\omega} \cdot \log M$$

SUMMARY OF RESULTS

$n^{2.575} \cdot M^{0.681}$	shortest paths
$\frac{n^{\omega}}{\epsilon} \cdot \log\left(\frac{W}{\epsilon}\right)$	almost shortest paths

WHAT NEXT ?

$n^{2.5}$	???
n^{ω}	???
$n^{3-\epsilon} \cdot \log M$	???

SUCCINCT SHORTEST PATHS