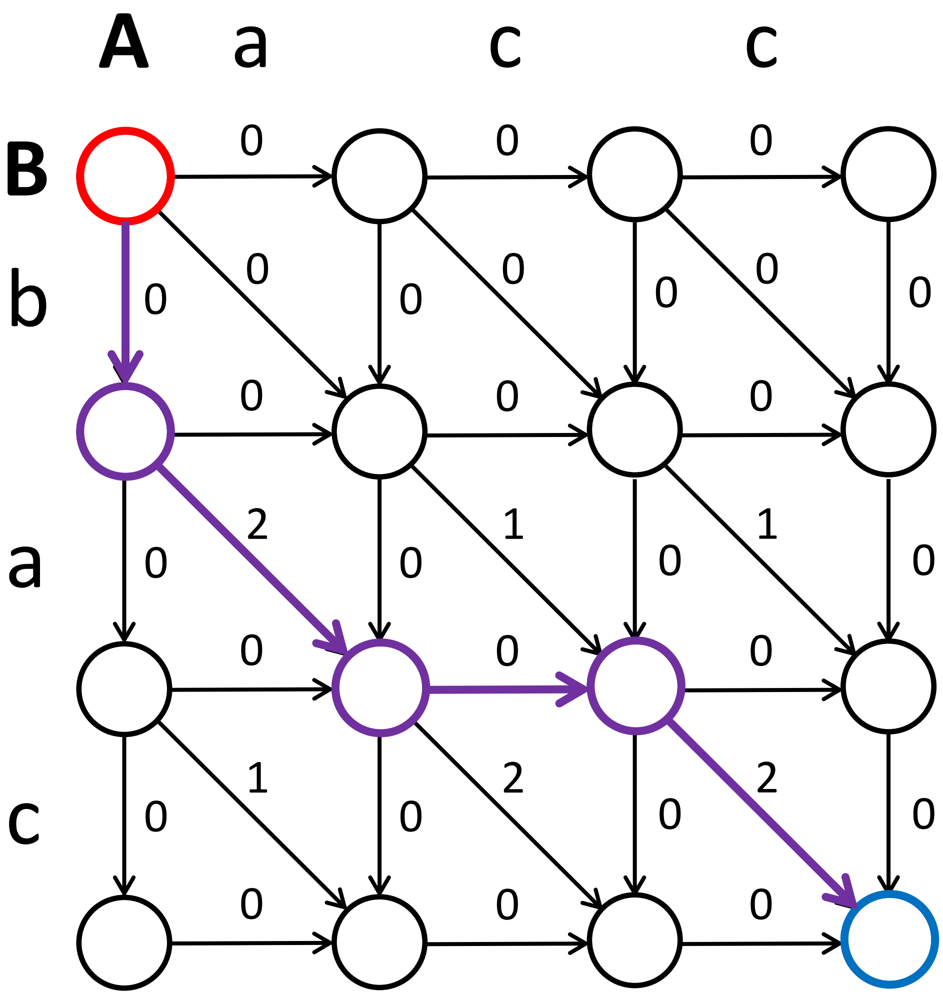
Efficient All Path Score Computations on Grid Graphs

Ury Matarazzo

Dekel Tsur

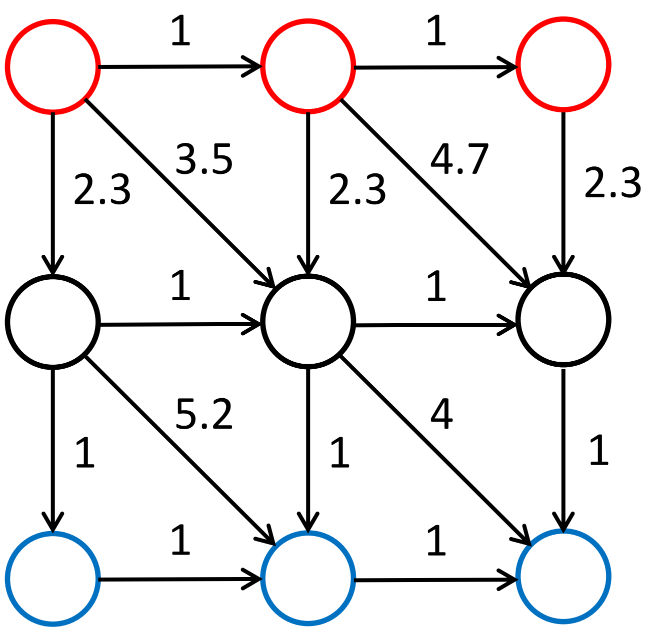
Michal Ziv-Ukelson

Grid graph from string alignment



δ	-	a	b	c
-		0	0	0
a	0	2	0	1
b	0	0	2	0
c	0	1	0	2

Grid All Path Scores (GAPS)

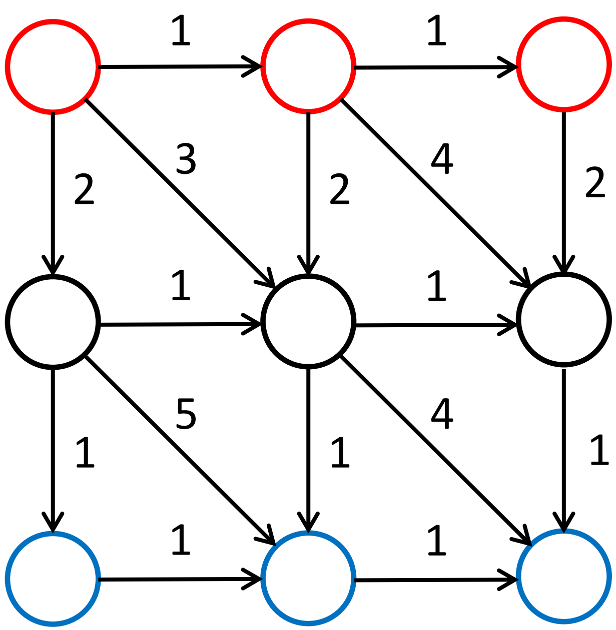


<u>Goal</u>

compute heaviest paths
from **every** vertex on first row
to **every** vertex on last row

Schmidt: $\boldsymbol{O(n^2 \cdot logn)}$

Integers Grid All Path Scores (IGAPS)

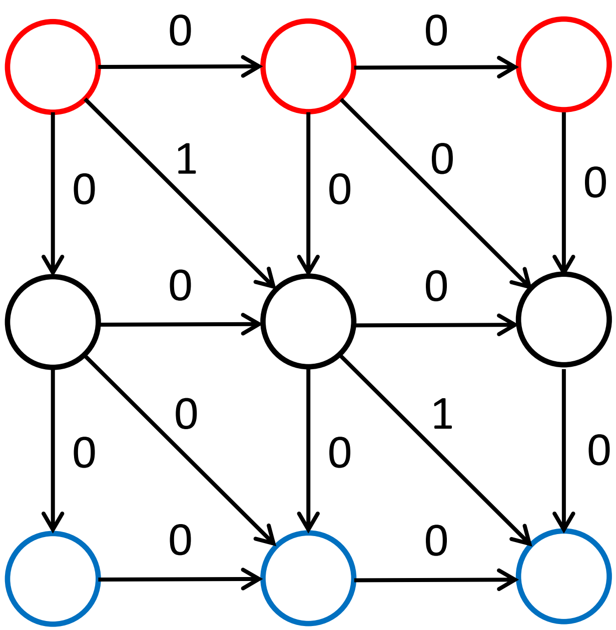


$C = 5$

Weights are integers from the set $\{0, 1, \ldots, C\}$

Schmidt: $\boldsymbol{O(C \cdot n^2)}$

Binary Grid All Path Scores (BGAPS)



Weights are integers from the set $\{0,1\}$

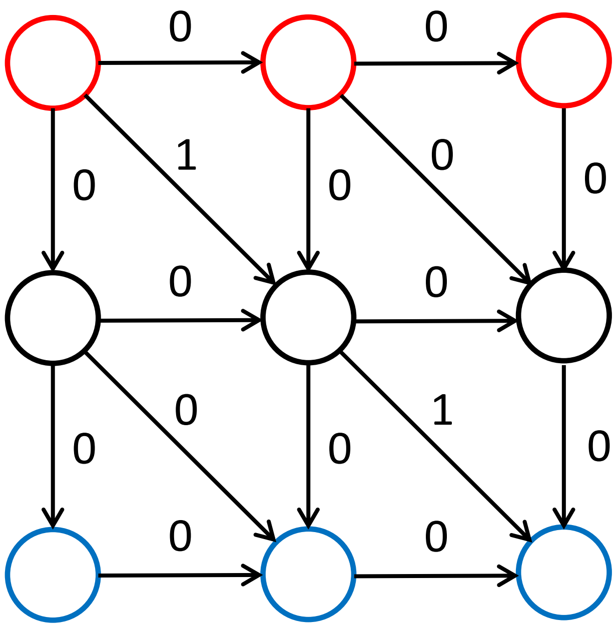
Schmidt: $O(n^2)$

Alves, Caceres, Song: $O(n^2)$

Tiskin: $O\left(n^2 \cdot (loglogn/logn)^2\right)$

For grid graph of LCS problem

Sparse Binary Grid All Path Scores (sparse BGAPS)



$r = 2$

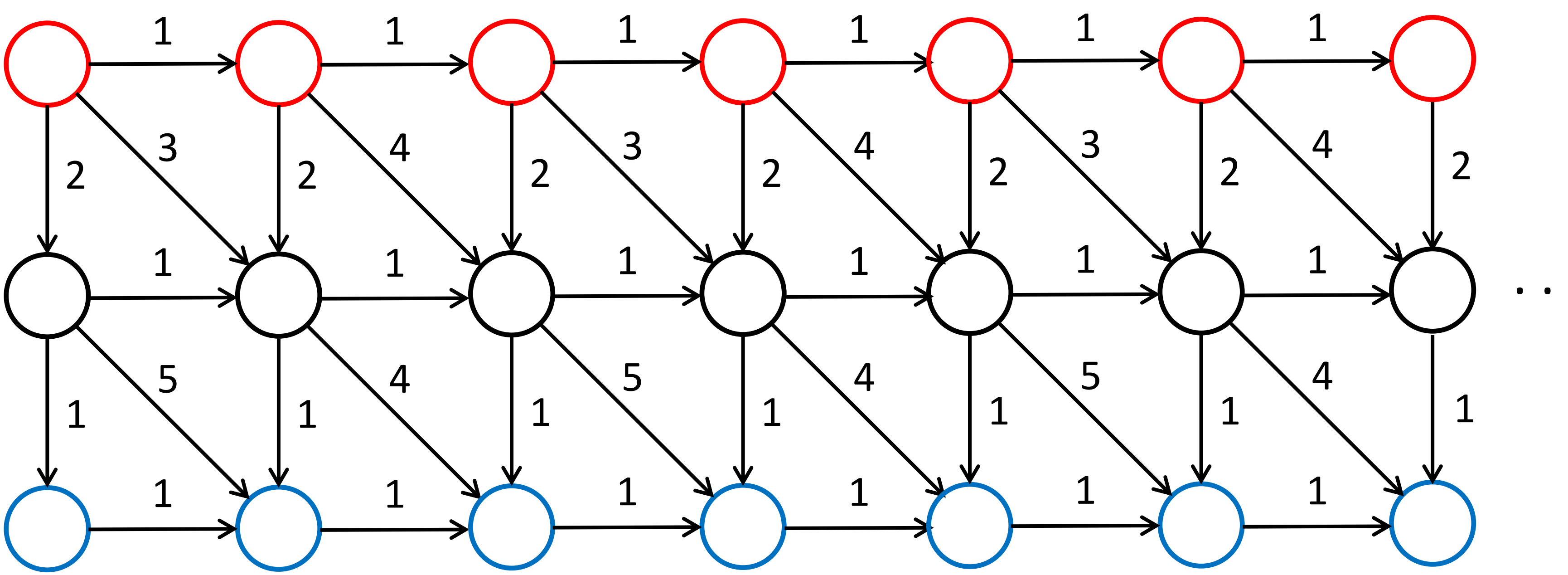
Horizontal and vertical weights are 0 and diagonal weights are 0 or 1. Number of diagonal edges of weights 1 is r.

Krusche, Tiskin: $\boldsymbol{O(n \cdot \sqrt{r})}$

Periodic Integers Grid All Path Scores (periodic IGAPS)

Weights are integers from the set $\{0,1,\ldots,C\}$. The graph is infinite and the weights values are periodic. The representation is finite.

Tiskin: $O(C^2 \cdot n^2)$



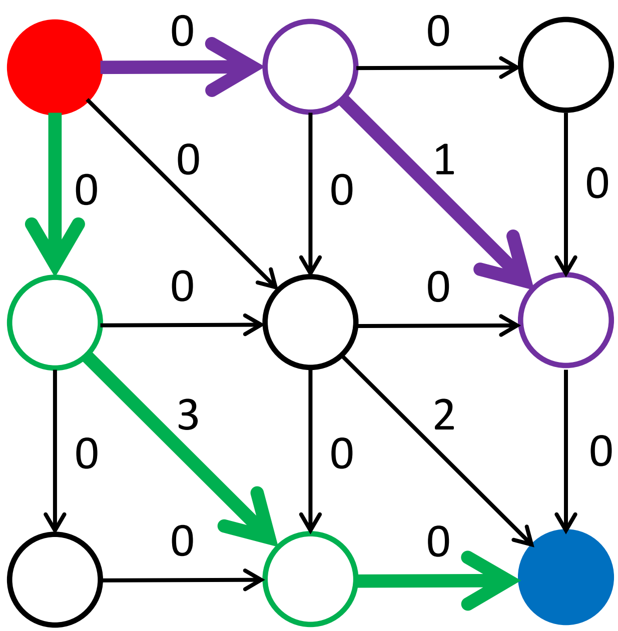
Contribution

- Integers Grid All Path Scores (IGAPS) - Weights are integers in {0,…,C}
- Binary Grid All Path Scores (BGAPS) - Weights are 0 or 1
- sparse BGAPS – r represents the amount of edges with weight 1
- periodic infinite – Graph is infinite and weights are periodic

	Finite	Infinite periodic
IGAPS	S, 1998: $C \cdot n^2$	T, 2009: $C^2 \cdot n^2$ $\boldsymbol{C \cdot n^2}$
IGAPS from alignment	T: $C^2 \cdot n^2 \cdot (\frac{loglogn}{logn})^2$	$\boldsymbol{C^2 \cdot n^2 \cdot (\frac{loglogn}{logn})^2}$
Sparse BGAPS	K, T, 2008: $n \cdot \sqrt{r}$ $\boldsymbol{r \cdot log^3(n^2/r)}$	$\boldsymbol{r \cdot log^3(n^2/r)}$

Some definitions

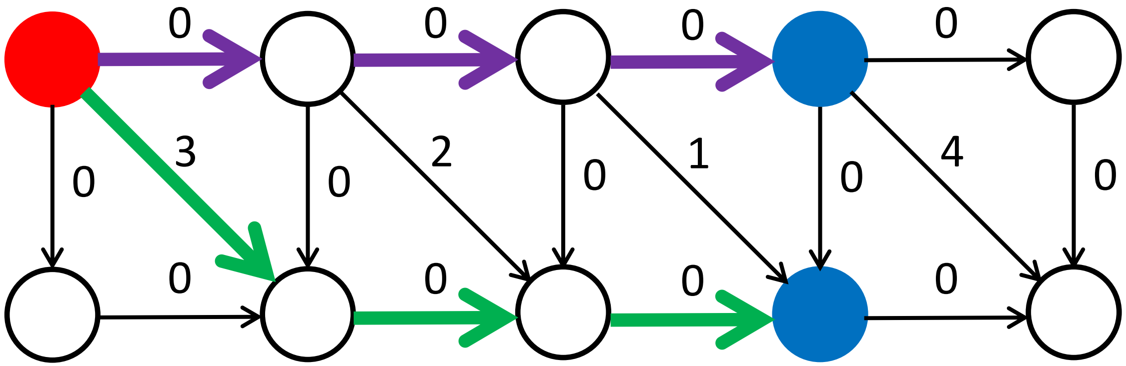
$OPT_i(k, j)$ is the maximum weight of a path from $(0, k)$ to (i, j)
DiffC $_{i,j}(k) = OPT_i(k, j + 1) - OPT_i(k, j)$
DiffR $_{i,j}(k) = OPT_{i+1}(k, j) - OPT_i(k, j)$



$OPT_2(0, 2) = 3$
$OPT_1(0, 2) = 1$
DiffR $_{1,2}(0) = 3 - 1 = 2$

We will compute all DiffR and DiffC values which give us implicit representation of the all-scores matrix of G.

Computing Diff with compact representation



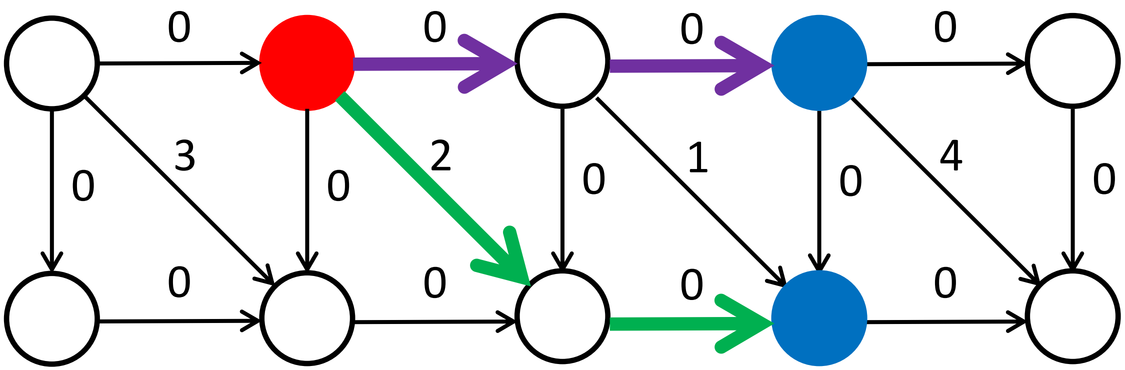
$OPT_0(0,3) = 0$

$\text{DiffR}_{0,3}(0) = 3$

$OPT_1(0,3) = 3$

$\text{DiffR}_{0,3} = (3,2,1,0)$

Computing Diff with compact representation



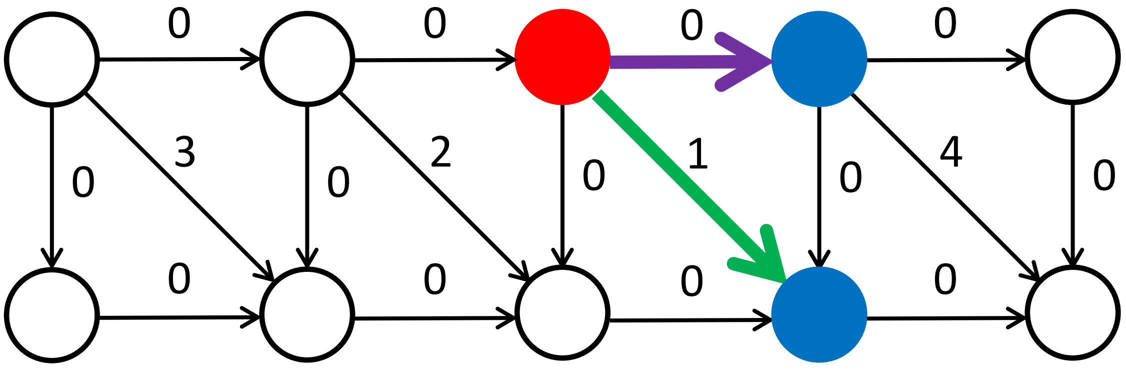
$OPT_0(1{,}3) = 0$

$\text{DiffR}_{0,3}(1) = 2$

$OPT_1(1{,}3) = 2$

$\text{DiffR}_{0,3} = (3{,}2{,}1{,}0)$

Computing Diff with compact representation



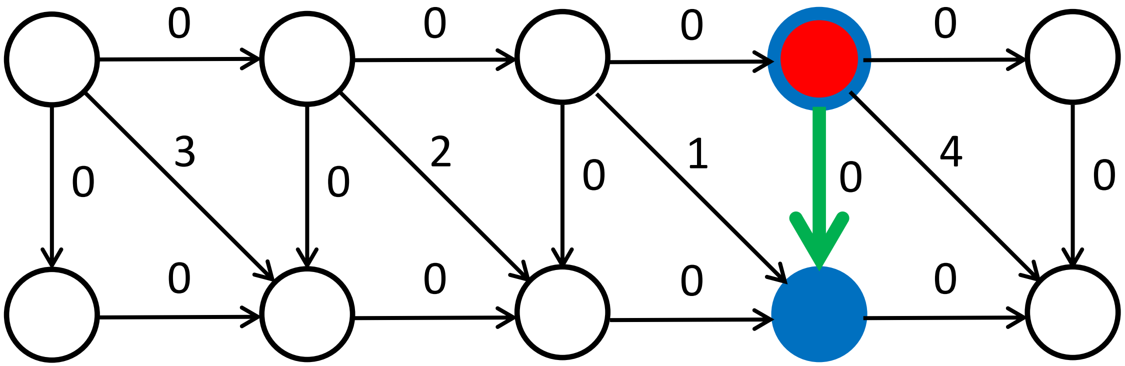
$OPT_0(2,3) = 0$

$\text{DiffR}_{0,3}(2) = 1$

$OPT_1(2,3) = 1$

$\text{DiffR}_{0,3} = (3,2,1,0)$

Computing Diff with compact representation



$$OPT_0(3,3) = 0$$

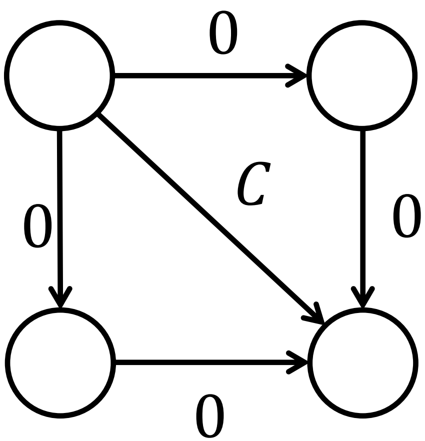
$$\text{DiffR}_{0,3}(3) = 0$$

$$OPT_1(3,3) = 0$$

$$\text{DiffR}_{0,3} = (3,2,1,0)$$

Bounding DiffC and DiffR

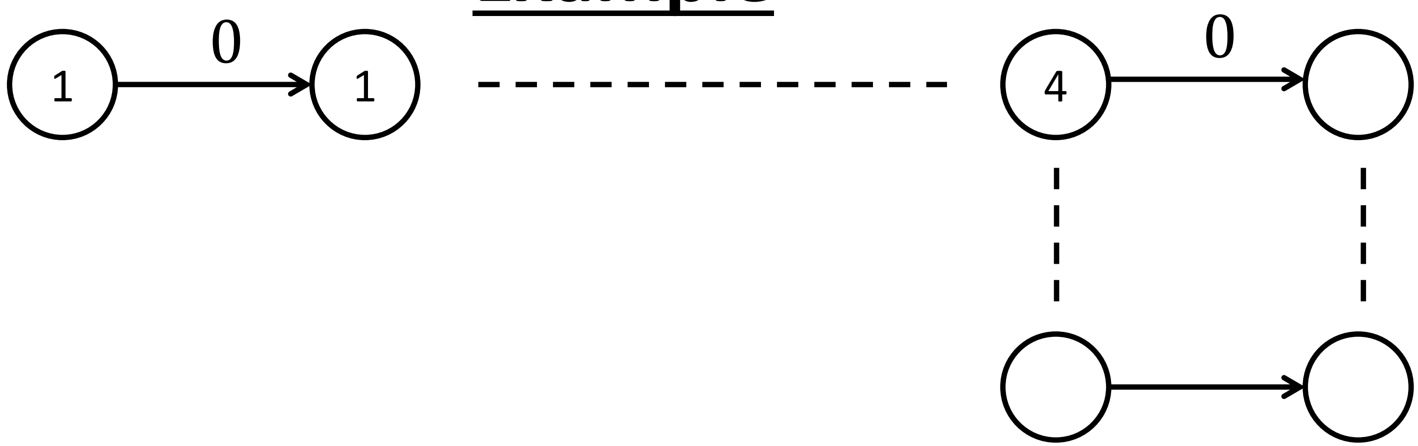
$0 \leq \text{DiffC}_{i,j}(k) \leq C$ and $0 \leq \text{DiffR}_{i,j}(k) \leq C$



Compact representation of Diff

$SC_{i,j}$ ($SR_{i,j}$) is sequence that contains the indices in which DiffC $_{i,j}$ (DiffR $_{i,j}$) function changes.

Example



$C = 5$

$n = 9$

$k = (1,2,3,4,5,6,7,8,9)$

$(-\infty, 6,9,9, \infty)$

DiffC $_{9,9} = (1,1,1,1,1,2,2,2,4)$ Diff function

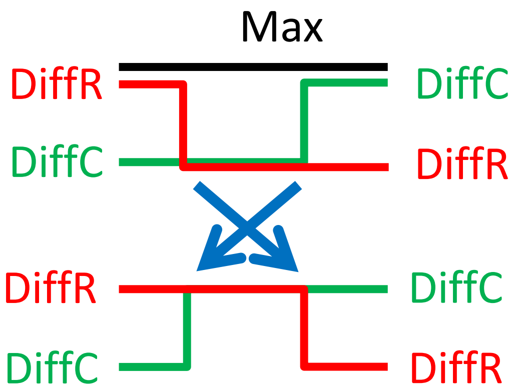
$SC_{i,j} = (-\infty, 6,9,9, \infty)$ Compact representation

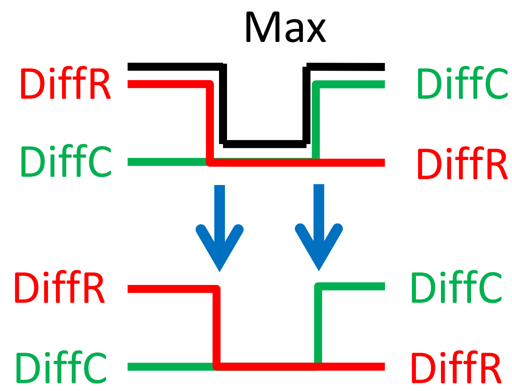
Contribution

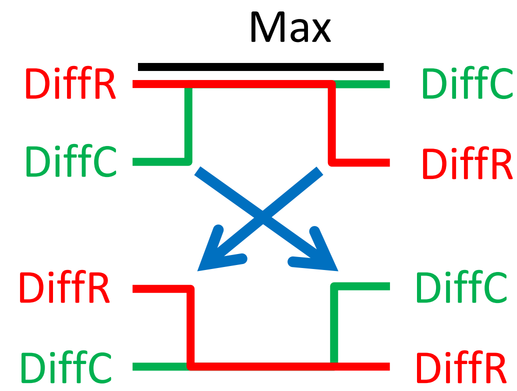
- Integers Grid All Path Scores (IGAPS) - Weights are integers in {0,...,C}
- Binary Grid All Path Scores (BGAPS) - Weights are 0 or 1
- sparse BGAPS – r represents the amount of edges with weight 1
- periodic infinite – Graph is infinite and weights are periodic

	Finite	Infinite periodic
IGAPS	S, 1998: $C \cdot n^2$	T, 2009: $C^2 \cdot n^2$ $\boldsymbol{C \cdot n^2}$
IGAPS from alignment	T: $C^2 \cdot n^2 \cdot (\frac{loglogn}{logn})^2$	$\boldsymbol{C^2 \cdot n^2 \cdot (\frac{loglogn}{logn})^2}$
Sparse BGAPS	K, T, 2008: $n \cdot \sqrt{r}$ $\boldsymbol{r \cdot log^3(n^2/r)}$	$\boldsymbol{r \cdot log^3(n^2/r)}$

Algorithm for finite sparse BGAPS







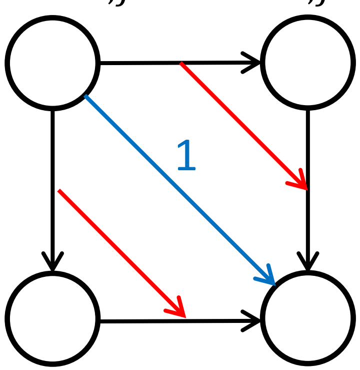
$SR < SC$

$SC < SR$

$\boldsymbol{W_{i,j} = 1}$

$SC_{i+1,j} = SR_{i,j}$

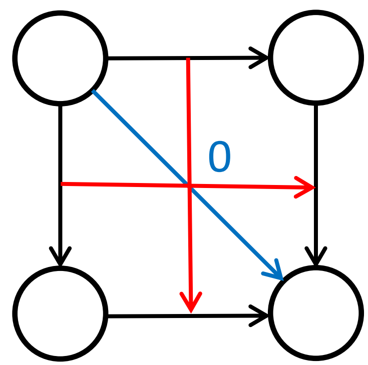
$SR_{i,j+1} = SC_{i,j}$



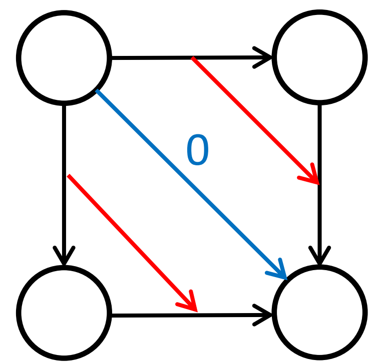
$\boldsymbol{W_{i,j} = 0}$

$SC_{i+1,j} = \max\{SC_{i,j}, SR_{i,j}\}$

$SR_{i,j+1} = \min\{SC_{i,j}, SR_{i,j}\}$

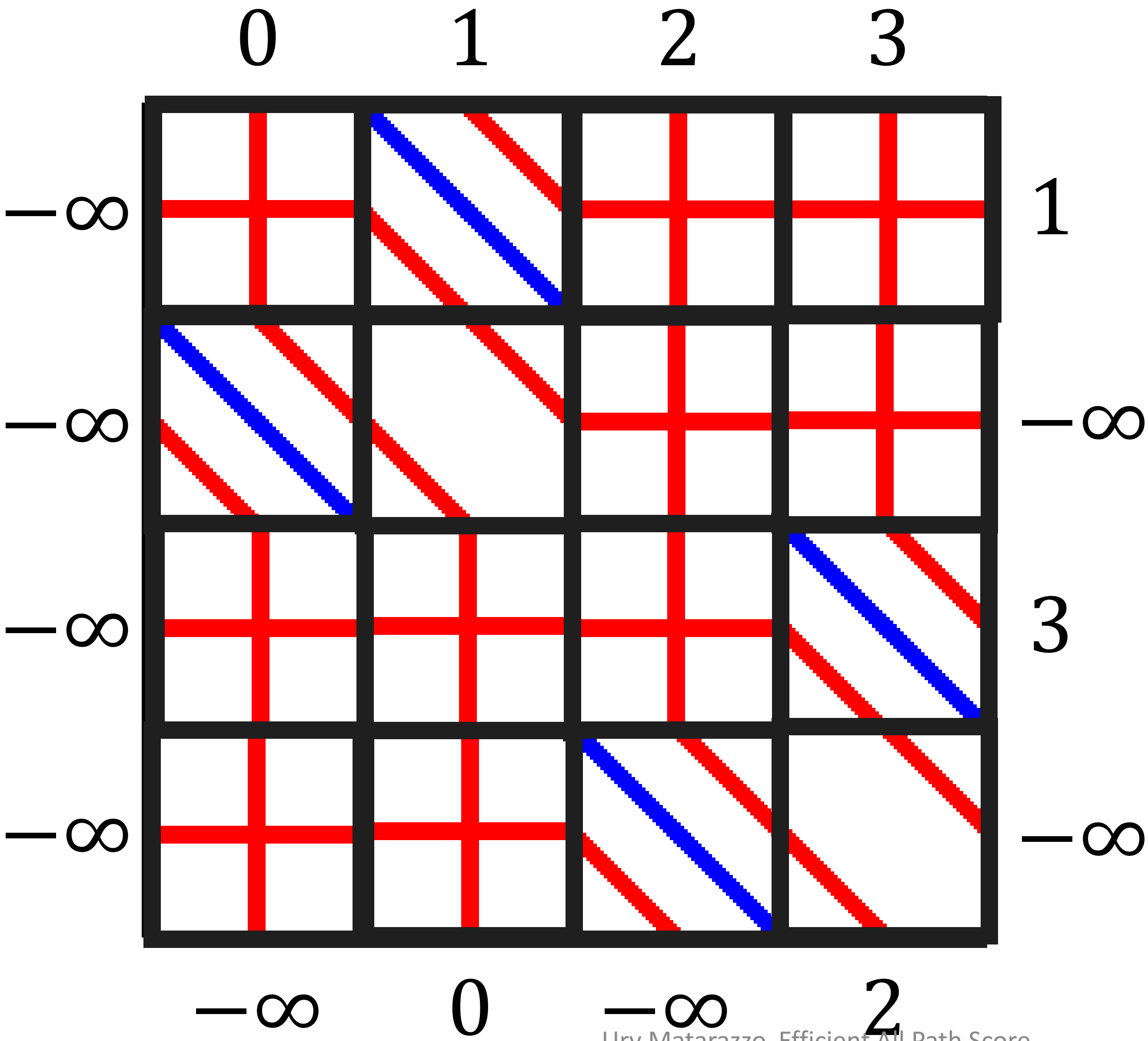


Indexes didn’t cross:
They cross once

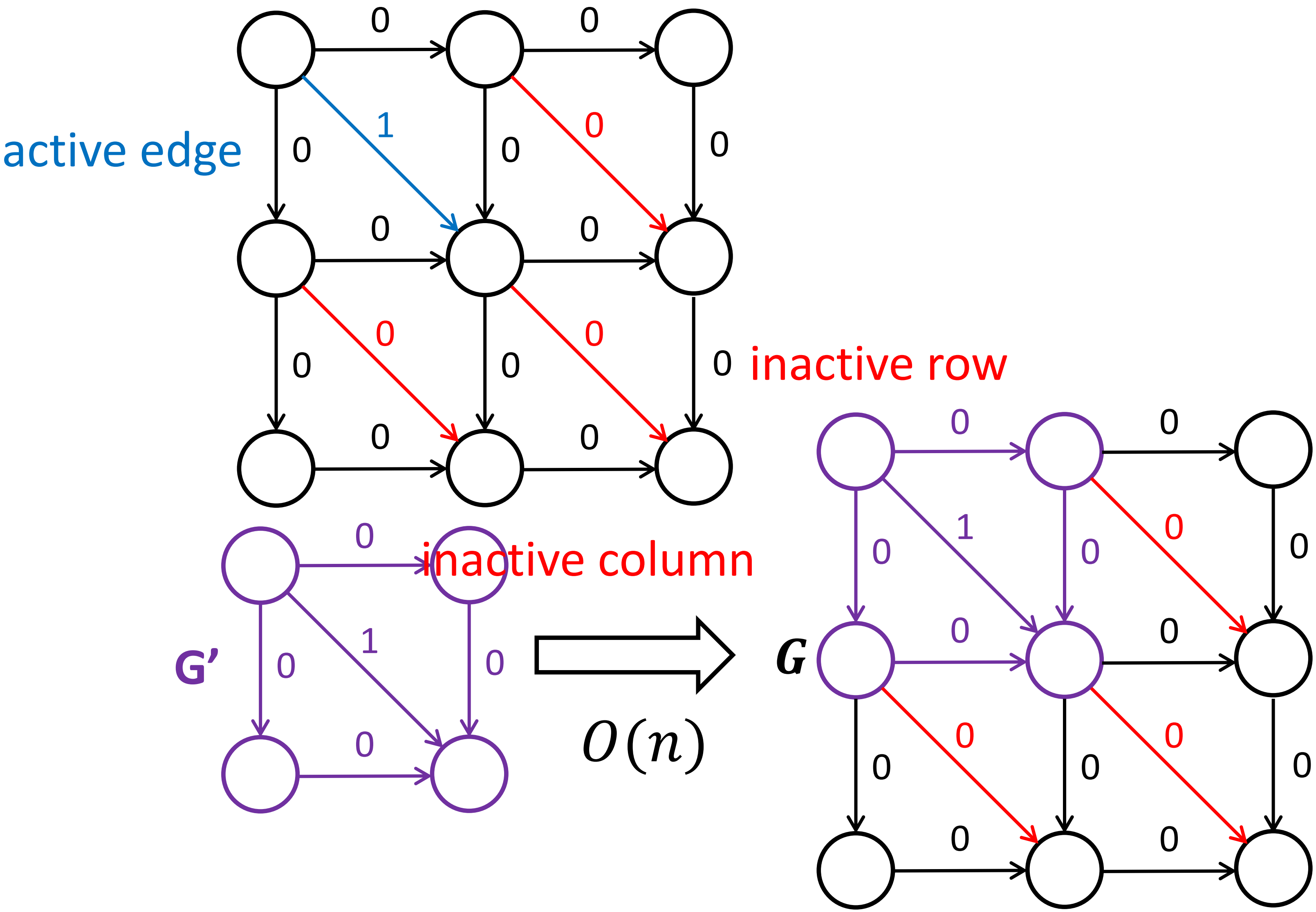


Indexes crossed:
don’t cross again

Algorithm for finite sparse BGAPS

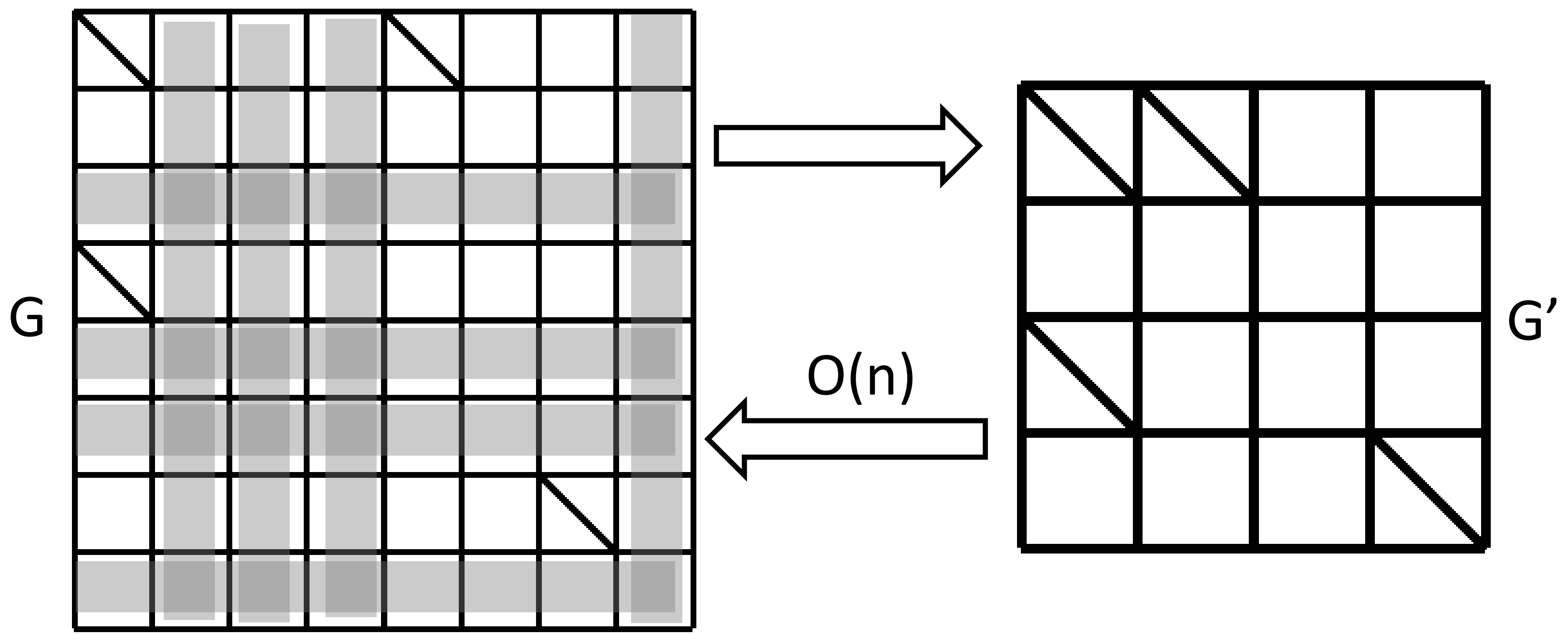


Algorithm for finite sparse BGAPS



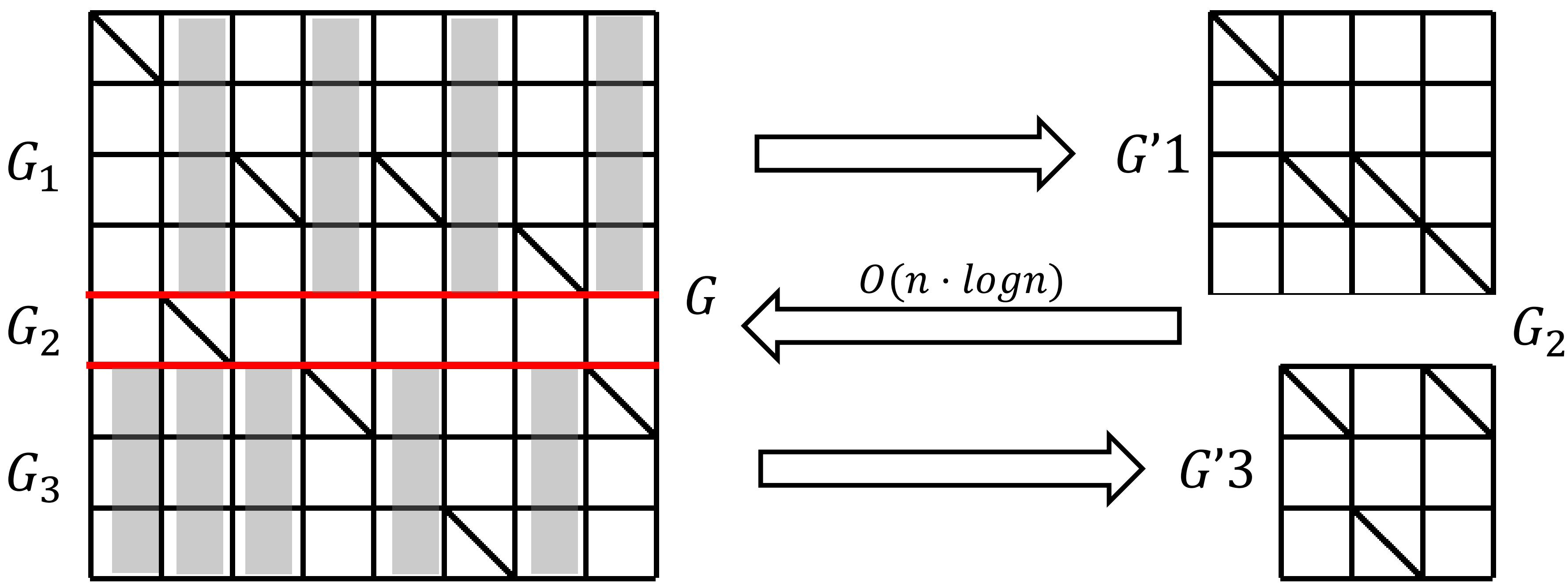
Algorithm for finite sparse BGAPS with $r \leq n$

- **Case 1:** G has at most n / 2 active edges:
- then there are at least n / 2 inactive rows and at least n / 2 inactive columns.
- Choose n / 2 inactive rows and n / 2 inactive columns and remove them from G to obtain a grid graph G'.
- Recursively compute $OUT(G')$ and then obtain $OUT(G)$.



Algorithm for finite sparse BGAPS with $r \leq n$

- **Case 2:** G has more than n / 2 active edges:
- let i be the maximum index such that $G_1 = G[0..i, 0..n]$ has at most n / 2 active edges.
- Let $G_2 = G[i..i+1, 0..n]$ and $G_3 = G[i+1..n, 0..n]$.
- G_1, G_2, G_3 have at most n / 2 active edges.

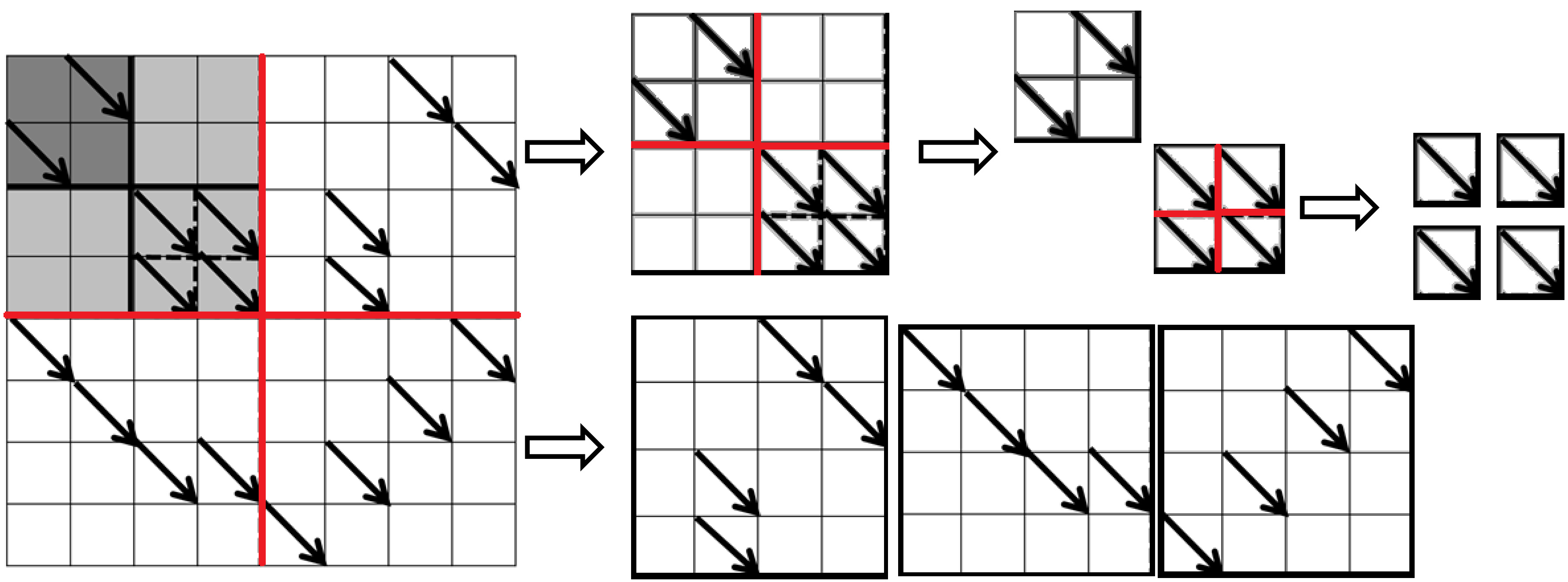


Algorithm for finite sparse BGAPS

- If the number of active edges is at most n, compute $OUT(G)$, Stop
- Partition G into four subgraphs:

  $G_1 = G[0..n/2, 0..n/2], G_2 = G[0..n/2, n/2..n]$,

  $G_3 = G[n/2..n, 0..n/2], and\ G_4 = G[n/2..n, n/2..n]$.
- Recursively compute $OUT(G_i)$ for each of the subgraphs.
- Compute $OUT(G)$ by application of Theorem 3 three times.

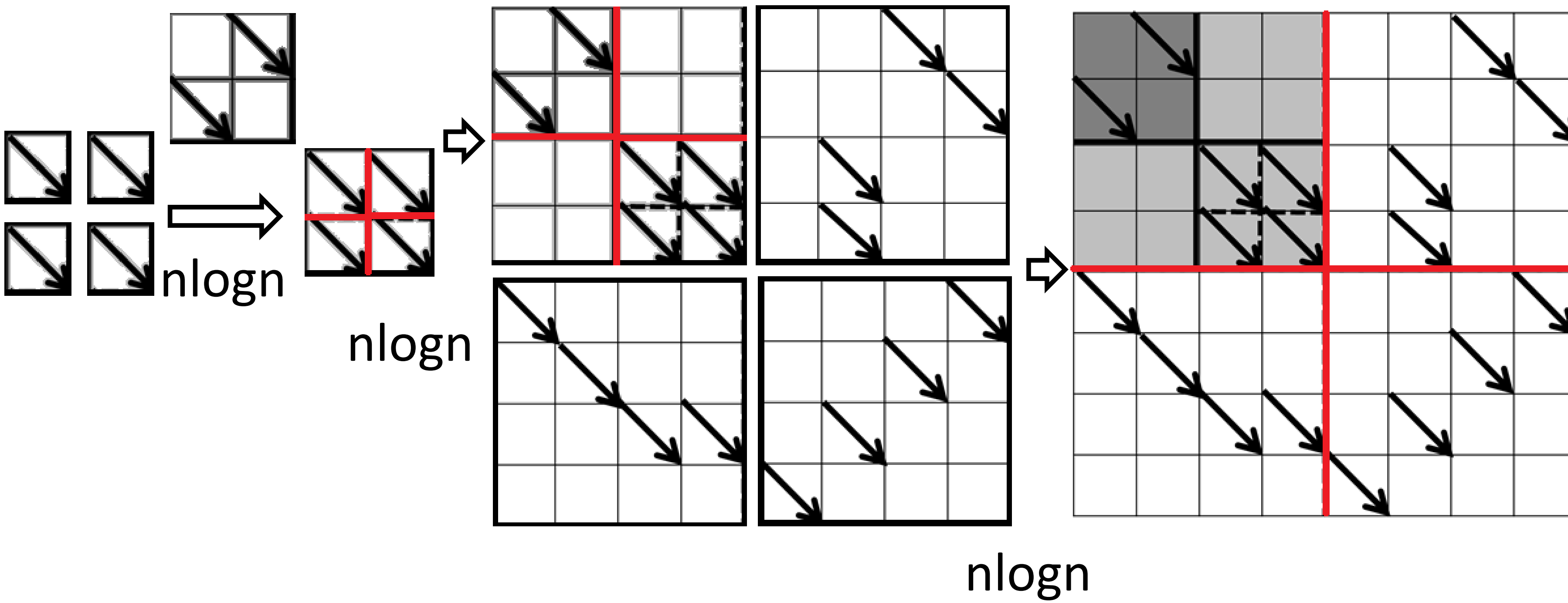


Algorithm for finite sparse BGAPS

- If the number of active edges is at most n, compute $OUT(G)$, Stop
- Partition G into four subgraphs:

  $G_1 = G[0..n/2, 0..n/2], G_2 = G[0..n/2, n/2..n],$

  $G_3 = G[n/2..n, 0..n/2], and\ G_4 = G[n/2..n, n/2..n].$
- Recursively compute $OUT(G_i)$ for each of the subgraphs.
- Compute $OUT(G)$ by application of Theorem 3 three times.



Time complexity

- number of graphs handled in level j is at most 4^j
- number of subgraphs in level $j-1$ with high r is at most $\frac{r}{n/2^{j-1}}$
- number of graphs handled in level j is at most $4 \times \frac{r}{n/2^{j-1}}$
- The time complexity of handling one subgraph in level j is either $O(n' \cdot log^2 n')$ if amount of active edges is smaller than n', and $O(n' \cdot log\, n')$ otherwise.
- Therefore, the total time of the algorithm is

$$O(\sum_{j=0}^{logn} \min(4^j, \frac{r \cdot 2^j}{n}) \cdot \frac{n}{2^j} log^2 \frac{n}{2^j}) = O(r \cdot log^3 \frac{n^2}{r})$$

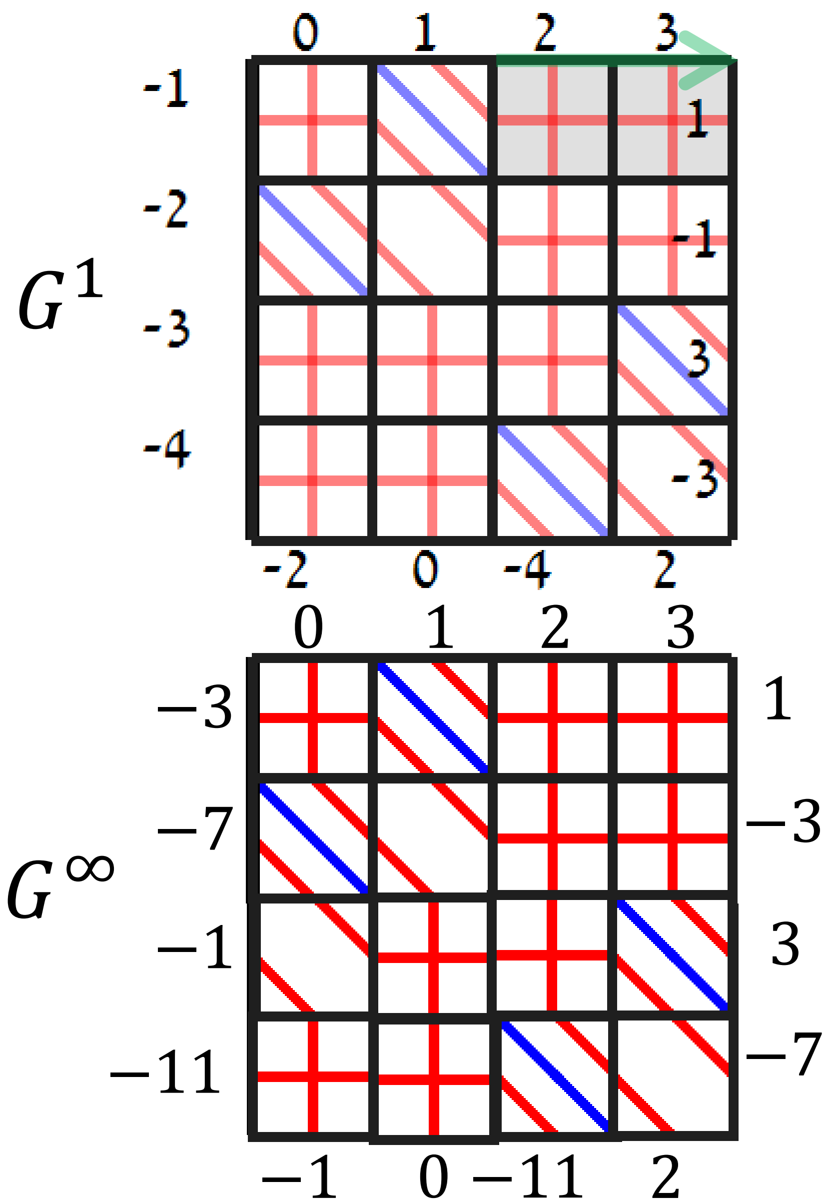
Where r is the number of edges whose weight is 1.

Contribution

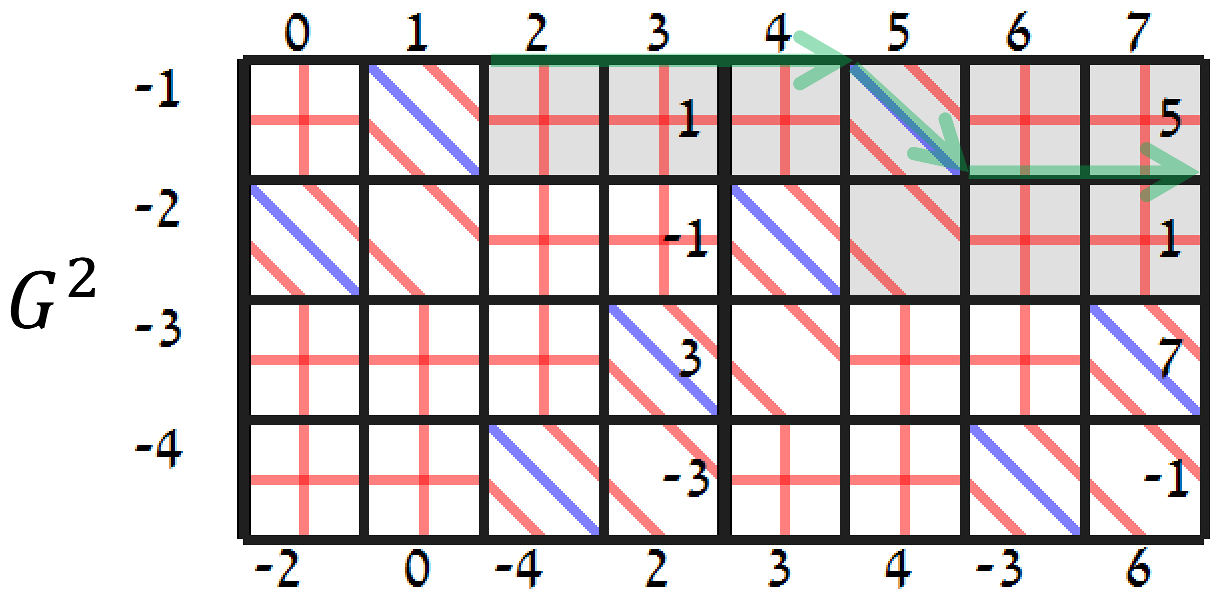
- Integers Grid All Path Scores (IGAPS) - Weights are integers in {0,…,C}
- Binary Grid All Path Scores (BGAPS) - Weights are 0 or 1
- sparse BGAPS – r represents the amount of edges with weight 1
- periodic infinite – Graph is infinite and weights are periodic

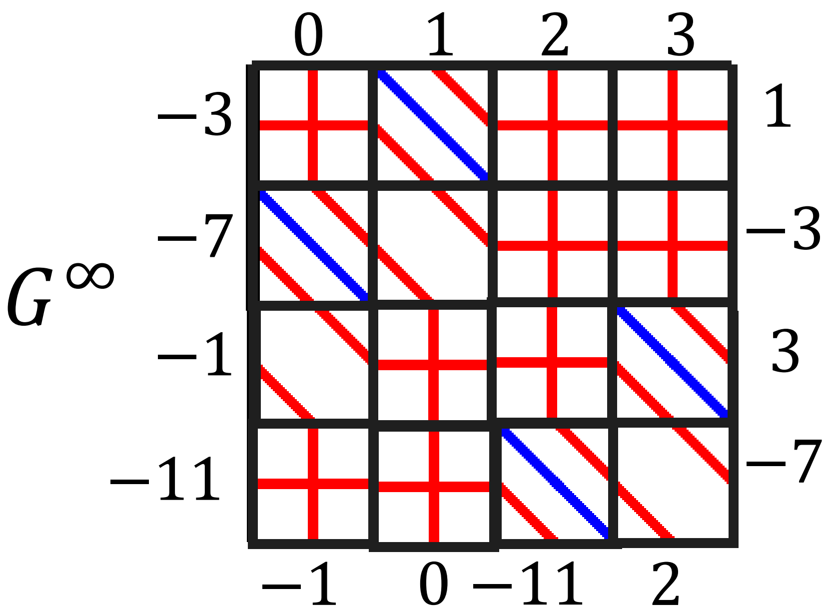
	Finite	Infinite periodic
IGAPS	S, 1998: $C \cdot n^2$	T, 2009: $C^2 \cdot n^2$ $\boldsymbol{C \cdot n^2}$
IGAPS from alignment	T: $C^2 \cdot n^2 \cdot (\frac{loglogn}{logn})^2$	$\boldsymbol{C^2 \cdot n^2 \cdot (\frac{loglogn}{logn})^2}$
Sparse BGAPS	K, T, 2008: $n \cdot \sqrt{r}$ $\boldsymbol{r \cdot log^3(n^2/r)}$	$\boldsymbol{r \cdot log^3(n^2/r)}$

Algorithm for periodic BGAPS



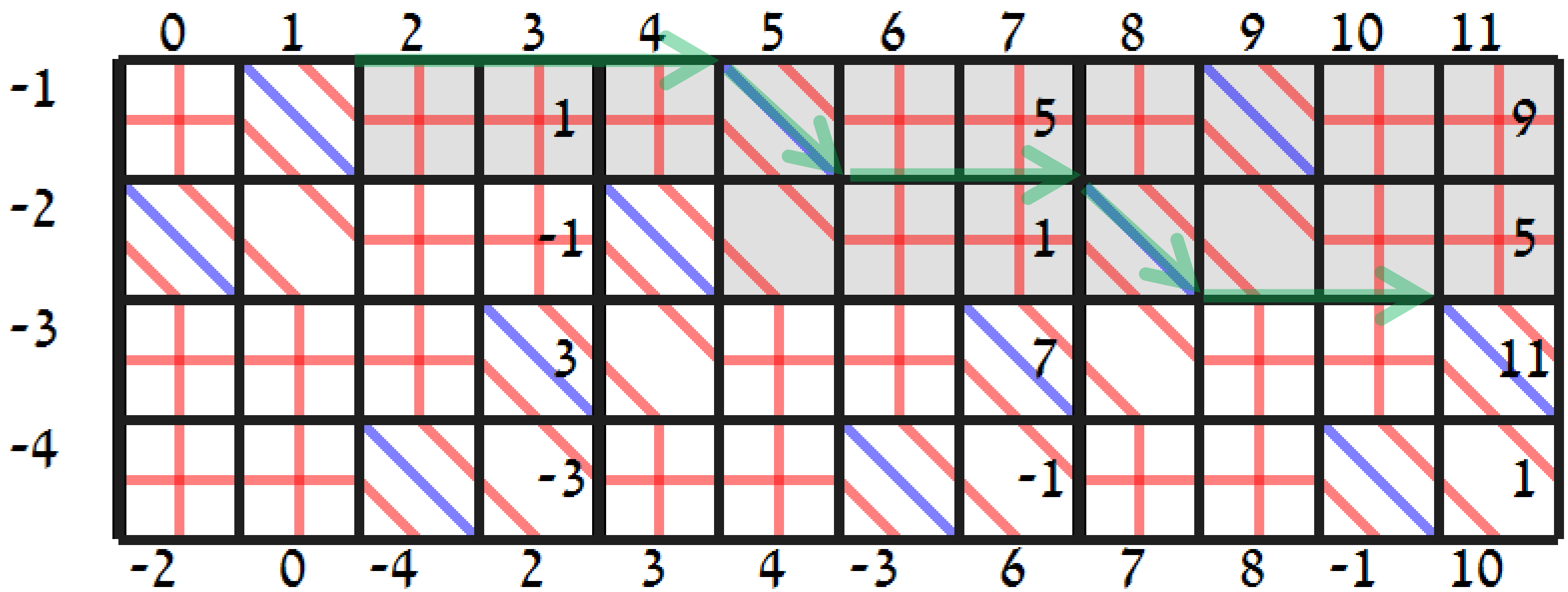
Algorithm for periodic BGAPS



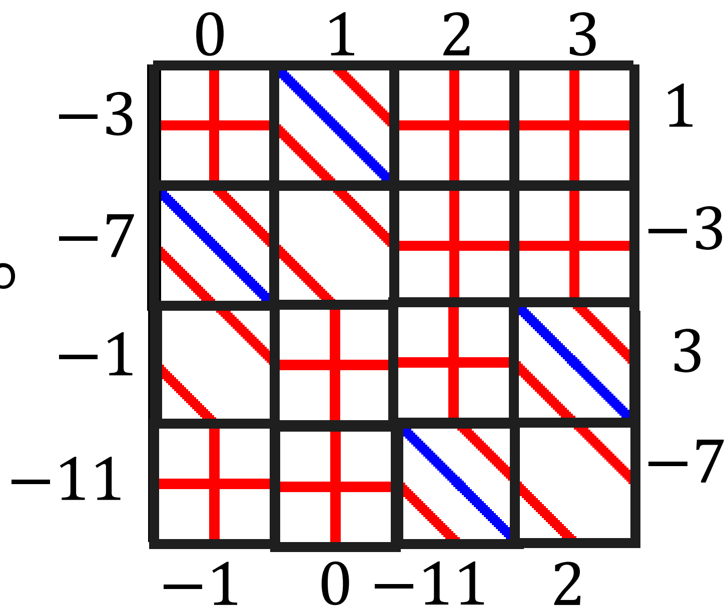


Algorithm for periodic BGAPS

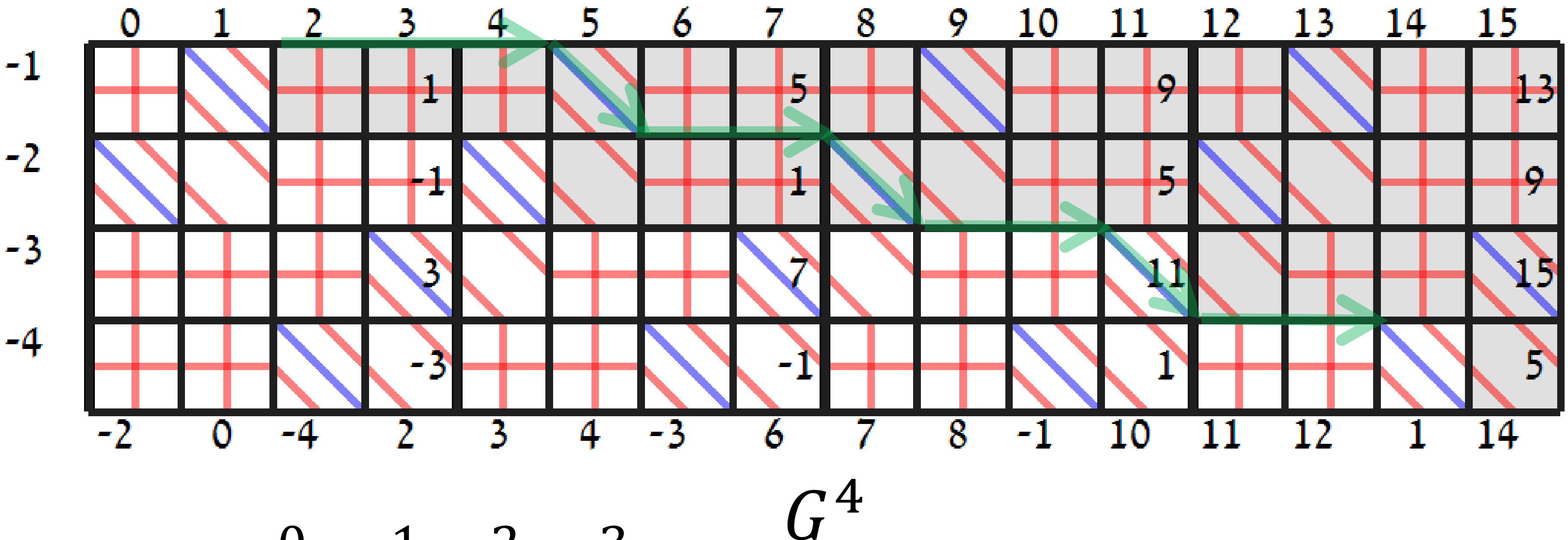
G^3

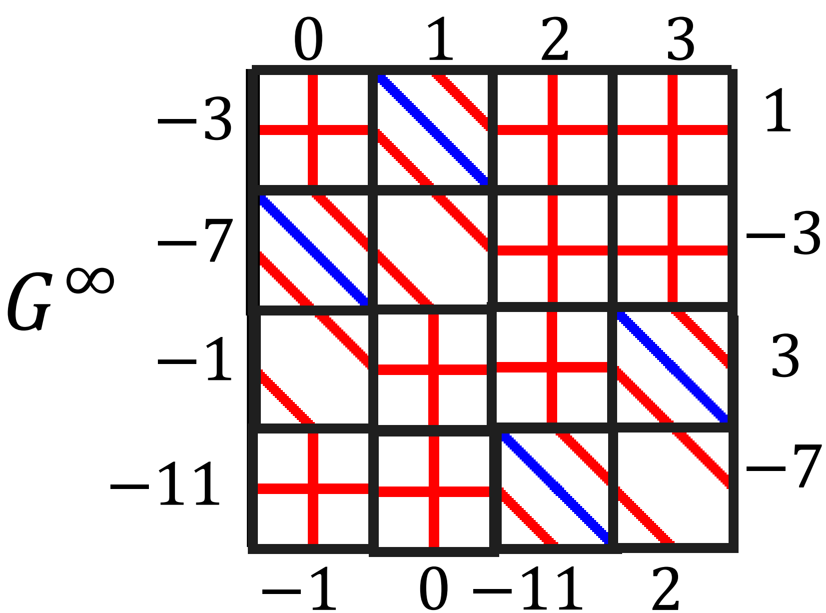


G^∞

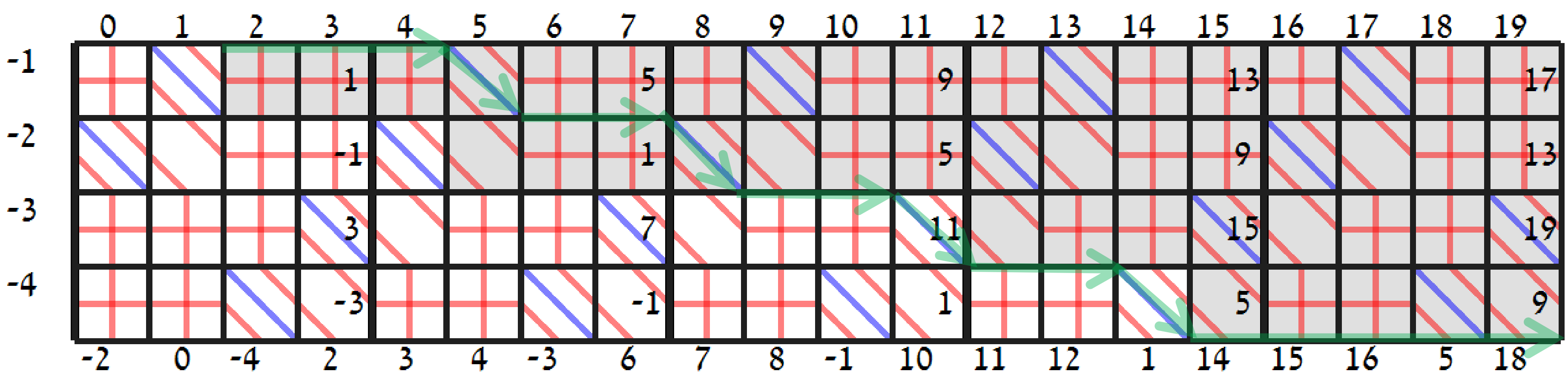


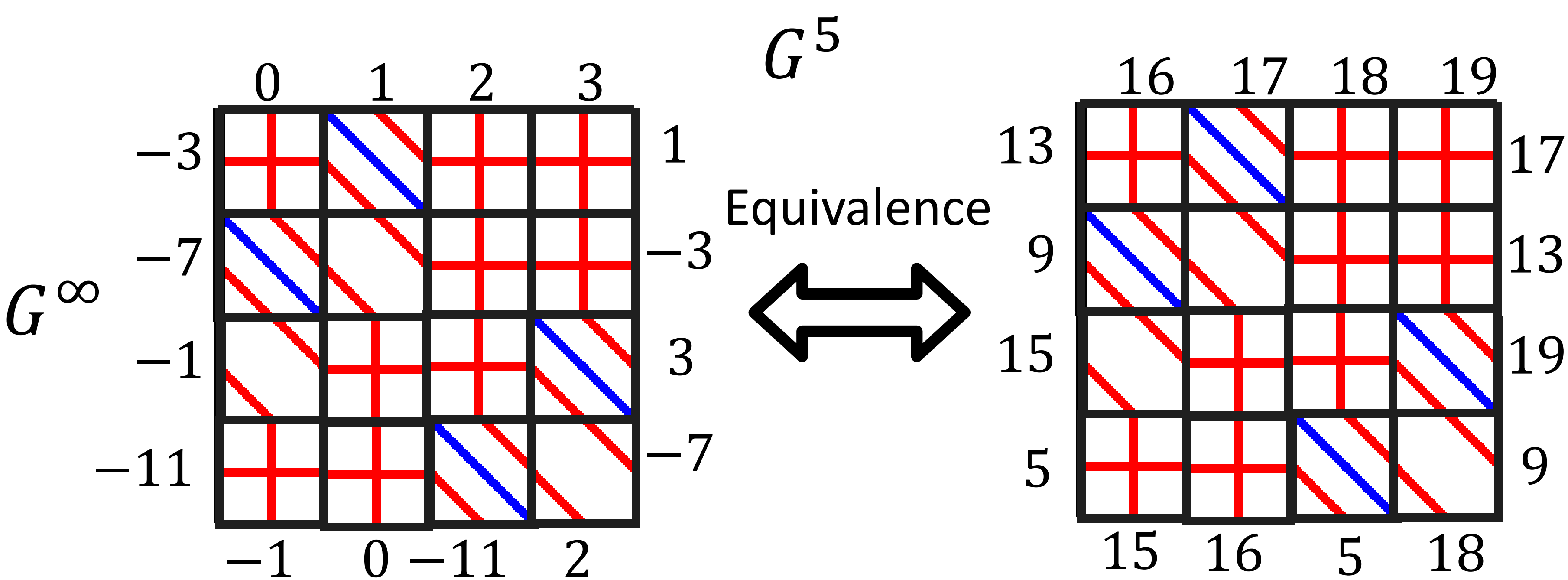
Algorithm for periodic BGAPS





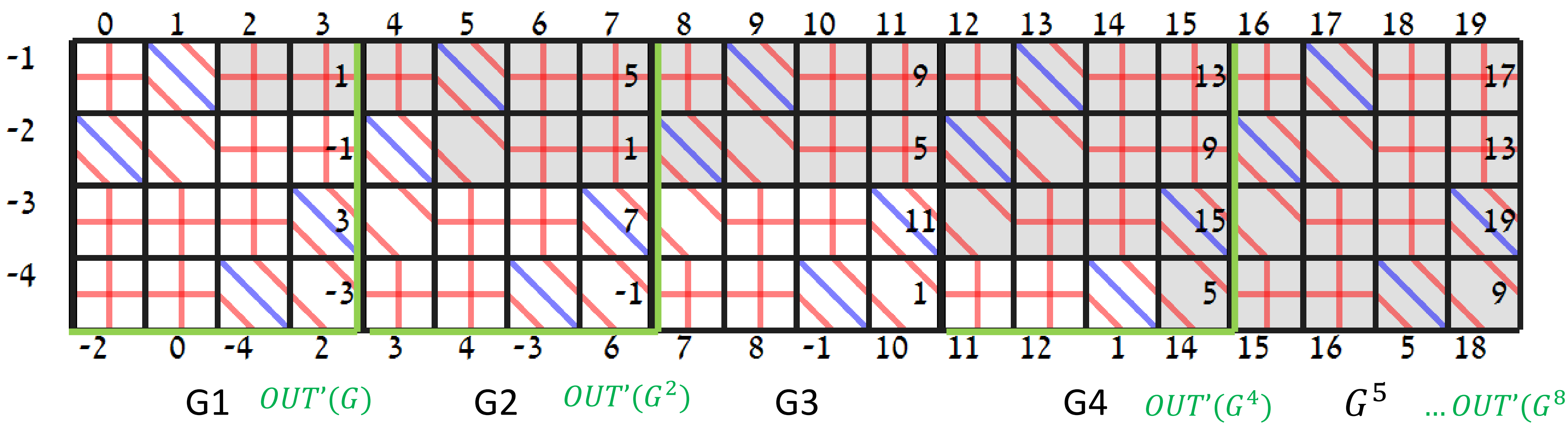
Algorithm for periodic BGAPS





Algorithm for periodic BGAPS

- (1) Compute $OUT'(G)$.
- (2) Let n' be the smallest power of 2 which is greater than or equal to n
- (3) For $k = 1, 2, 4, \dots, n' / 2$ do
- (4) Compute $OUT'\left(G^{2k}\right)$ from $OUT'\left(G^{k}\right)$
- (5) Output $(OUT'(G^{n'})[1] - (n'-1)n, \dots, OUT'(G^{n'})[n] - (n'-1)n)$



- Runtime: $T(G) + O(n \cdot log^2 n)$, where
- $T(G)$ is the time complexity of solving BGAPS on G.

Summary

- sparse BGAPS – r represents the amount of edges with weight 1
- periodic sparse BGAPS – Graph is infinite and weights are periodic

	Finite	Infinite periodic
IGAPS	S, 1998: $C \cdot n^2$	T, 2009: $C^2 \cdot n^2$ $\boldsymbol{C \cdot n^2}$
IGAPS from alignment	T: $C^2 \cdot n^2 \cdot (\frac{loglogn}{logn})^2$	$\boldsymbol{C^2 \cdot n^2 \cdot (\frac{loglogn}{logn})^2}$
Sparse BGAPS	K, T, 2008: $n \cdot \sqrt{r}$ $\boldsymbol{r \cdot log^3(n^2/r)}$	$\boldsymbol{r \cdot log^3(n^2/r)}$

Thank you