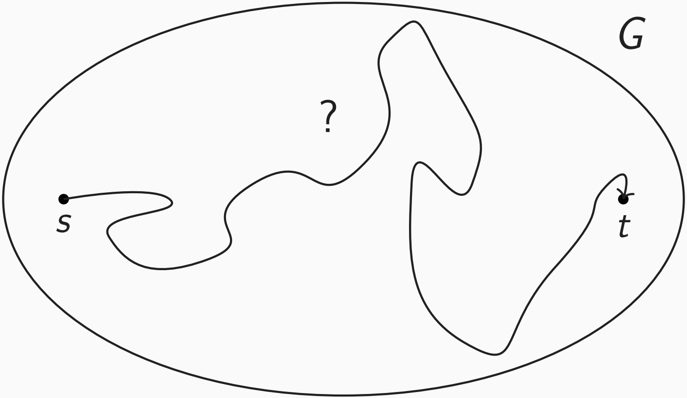
Frontier Space-Time Algorithms Using Only Full Memory

Petr Chmel[1], Aditi Dudeja[2], Michal Koucký[1], Ian Mertz[1], Ninad Rajgopal[1]
[1]: Charles University
[2]: The Chinese University of Hong Kong, Shenzhen

The problem

STCONN: given a directed graph G and $s, t \in V(G)$, decide whether there is a path from s to t.



n = number of vertices, m = number of edges

What do we know about STCONN

- BFS: $\tilde{\mathcal{O}}(n)$ space and $\mathcal{O}(n+m)$ time

What do we know about STCONN

- BFS: $\tilde{\mathcal{O}}(n)$ space and $\mathcal{O}(n+m)$ time
- Savitch'70: $\mathcal{O}(\log^2(n))$ space

What do we know about STCONN

- BFS: $\tilde{\mathcal{O}}(n)$ space and $\mathcal{O}(n+m)$ time
- Savitch'70: $\mathcal{O}(\log^2(n))$ space and $n^{\Theta(\log n)}$ time

What do we know about STCONN

- BFS: $\tilde{\mathcal{O}}(n)$ space and $\mathcal{O}(n+m)$ time
- Savitch'70: $\mathcal{O}(\log^2(n))$ space and $n^{\Theta(\log n)}$ time
- Many results about restricted classes

What do we know about STCONN

- BFS: $\tilde{\mathcal{O}}(n)$ space and $\mathcal{O}(n+m)$ time
- Savitch'70: $\mathcal{O}(\log^2(n))$ space and $n^{\Theta(\log n)}$ time
- Many results about restricted classes
  - e.g., Reingold'04: USTCONN $\in \mathrm{L}$

What do we know about STCONN

- BFS: $\tilde{\mathcal{O}}(n)$ space and $\mathcal{O}(n+m)$ time
- Savitch'70: $\mathcal{O}(\log^2(n))$ space and $n^{\Theta(\log n)}$ time
- Many results about restricted classes
  - e.g., Reingold'04: USTCONN $\in \mathrm{L}$
- What about sublinear space *and polynomial time*?

What do we know about STCONN

- BFS: $\tilde{\mathcal{O}}(n)$ space and $\mathcal{O}(n+m)$ time
- Savitch'70: $\mathcal{O}(\log^2(n))$ space and $n^{\Theta(\log n)}$ time
- Many results about restricted classes
  - e.g., Reingold'04: USTCONN $\in \mathrm{L}$
- What about sublinear space *and polynomial time*?
- Barnes, Buss, Ruzzo, Schieber'98: $n/2^{\Omega(\sqrt{\log n})}$ space and poly(n) time

What do we know about STCONN

- BFS: $\tilde{\mathcal{O}}(n)$ space and $\mathcal{O}(n+m)$ time
- Savitch'70: $\mathcal{O}(\log^2(n))$ space and $n^{\Theta(\log n)}$ time
- Many results about restricted classes
  - e.g., Reingold'04: USTCONN $\in \mathrm{L}$
- What about sublinear space ***and polynomial time***?
- Barnes, Buss, Ruzzo, Schieber'98: $n/2^{\Omega(\sqrt{\log n})}$ space and poly(n) time
- Edmonds, Poon, Achlioptas'99: matching space lower bound for poly(n)-time for pebbling algorithms (NNJAG)

Catalytic space

- Can we use space *for both storage and computation* at the same time?

Catalytic space

- Can we use space *for both storage and computation* at the same time?
- Useful for bypassing pebbling lower bounds

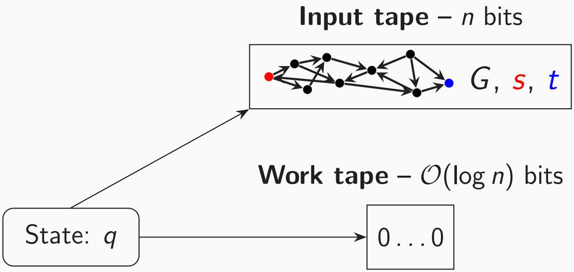
Catalytic space

- Can we use space *for both storage and computation* at the same time?
- Useful for bypassing pebbling lower bounds
- Cook, Mertz'24: Tree Evaluation in $\mathcal{O}(\log n \cdot \log \log n)$ space

Catalytic space

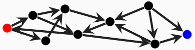
- Can we use space *for both storage and computation* at the same time?
- Useful for bypassing pebbling lower bounds
- Cook, Mertz'24: Tree Evaluation in $\mathcal{O}(\log n \cdot \log\log n)$ space
- Williams'25 using Cook, Mertz'24: For multitape Turing machines, $\mathrm{TIME}[t] \subseteq \mathrm{SPACE}[\sqrt{t \log t}]$

Catalytic Turing machine [BCKLS'14]



Catalytic Turing machine [BCKLS'14]

Input tape – n bits



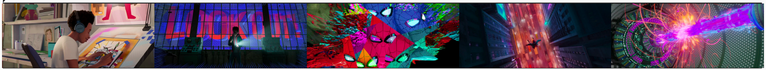
G, s, t

Work tape – $\mathcal{O}(\log n)$ bits

State: q

$0 \ldots 0$

Catalytic tape – poly(n) bits



Catalytic logspace

What can be solved in catalytic logspace?

Catalytic logspace

What can be solved in catalytic logspace?

- BCKLS'14: Logspace uniform TC^1 – STCONN, DET, ...

Catalytic logspace

What can be solved in catalytic logspace?

- BCKLS'14: Logspace uniform TC^1 – STCONN, DET, ...
- Chakraborty, Datta, Kusre, Mukhopadhyay, Sinhababu'26: Matching in general graphs

Catalytic logspace

What can be solved in catalytic logspace?

- BCKLS'14: Logspace uniform TC^1 – STCONN, DET, ...
- Chakraborty, Datta, Kusre, Mukhopadhyay, Sinhababu'26: Matching in general graphs
- Agarwala, Alekseev, Vinciguerra'26: Linear matroid intersection

Our results

Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\mathrm{poly}(n)$ time.

Our results

Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\text{poly}(n)$ time.

	Free space	Catalytic space	Time
BCKLS'14	$\mathcal{O}(\log n)$	$\text{poly}(n)$	$\text{poly}(n)$
Cook, Pyne'26	$\mathcal{O}(\log n)$	$\tilde{\mathcal{O}}(n^2)$	$\tilde{\mathcal{O}}(n^3 m)$
Theorem 1	$\mathcal{O}(\log n)$	$n/2^{\Omega(\sqrt{\log n})}$	$\text{poly}(n)$

$n =$ number of vertices, $m =$ number of edges

Our results, randomized

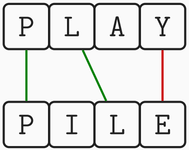
Theorem $1+\varepsilon$
STCONN can be solved with a randomized algorithm in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\mathcal{O}(n^{2+\varepsilon}m)$ time.

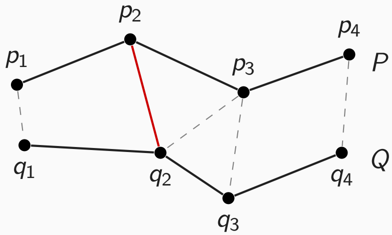
	Free space	Catalytic space	Time
Cook, Pyne'26	$\mathcal{O}(\log n)$	$\tilde{\mathcal{O}}(n)$	$\tilde{\mathcal{O}}(nm)$
Theorem $1+\varepsilon$	$\mathcal{O}(\log n)$	$n/2^{\Omega(\sqrt{\log n})}$	$\mathcal{O}(n^{2+\varepsilon}m)$

n = number of vertices, m = number of edges

Other problems

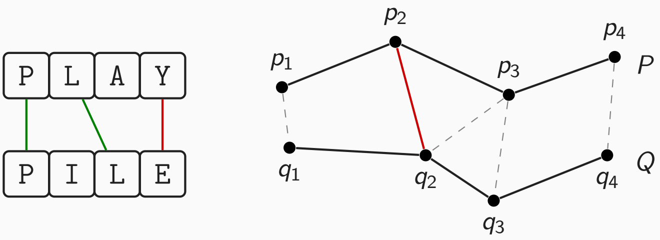
- Three other problems:
  - Edit distance (ED)
  - Longest common subsequence (LCS)
  - Discrete Fréchet distance (DFD)





Other problems

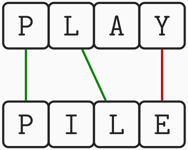
- Three other problems:
  - Edit distance (ED)
  - Longest common subsequence (LCS)
  - Discrete Fréchet distance (DFD)

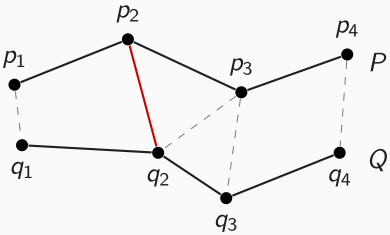


- All reduce to finding (weighted) longest/shortest/any path in a “grid” graph

Other problems

- Three other problems:
  - Edit distance (ED)
  - Longest common subsequence (LCS)
  - Discrete Fréchet distance (DFD)

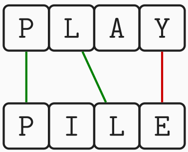


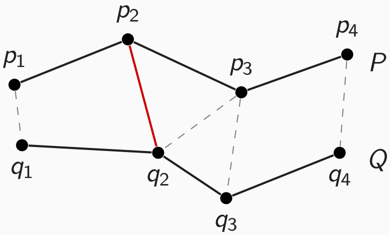


- All reduce to finding (weighted) longest/shortest/any path in a “grid” graph – $\Theta(n^2)$ *blowup*

Other problems

- Three other problems:
  - Edit distance (ED)
  - Longest common subsequence (LCS)
  - Discrete Fréchet distance (DFD)





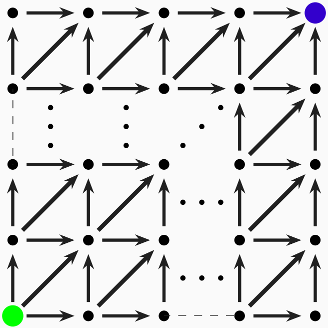
- All reduce to finding (weighted) longest/shortest/any path in a "grid" graph – $\Theta(n^2)$ *blowup*
- Kiyomi, Horiyama, Otachi'21: $n/2^{\Omega(\sqrt{\log n})}$-space poly-time algorithms for ED and LCS.

Solving ED, LCS, DFD

Theorem 2
ED, LCS and DFD can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\mathrm{poly}(n)$ time.

Theorem 2

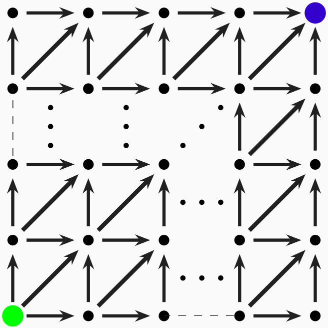
ED, LCS and DFD can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and poly(n) time.



- The graph has $\Theta(n^2)$ vertices

Theorem 2

ED, LCS and DFD can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\mathrm{poly}(n)$ time.



- The graph has $\Theta(n^2)$ vertices
- Need to compute the *shortest* path

BBRS: Long paths

Would like to use **BFS**

BBRS: Long paths

Would like to use **BFS** – need a smaller graph

BBRS: Long paths

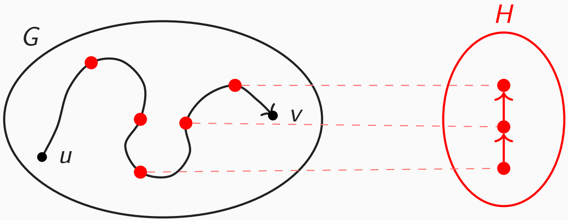
Would like to use **BFS** – need a smaller graph

Graph on n/λ vertices $\rightsquigarrow$ space $\mathcal{O}(n/\lambda)$

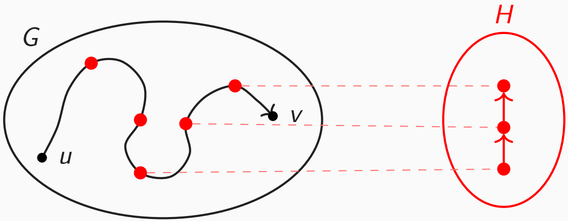
BBRS: Long paths

Would like to use **BFS** – need a smaller graph

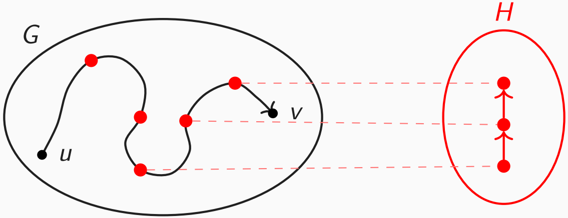
Graph on n/λ vertices $\rightsquigarrow$ space $\mathcal{O}(n/\lambda)$



BBRS: Sparsification



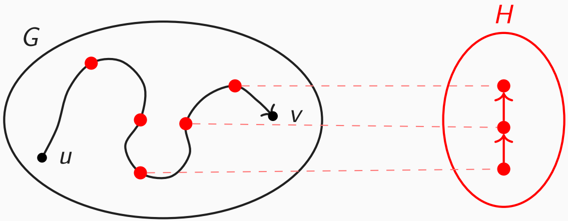
BBRS: Sparsification



Choose $\lambda := 2^{\Theta(\sqrt{\log n})}$, find a set S of n/λ vertices such that

- s, v connected in $G \Rightarrow \exists\ s, v$-path that hits S every λ steps

BBRS: Sparsification

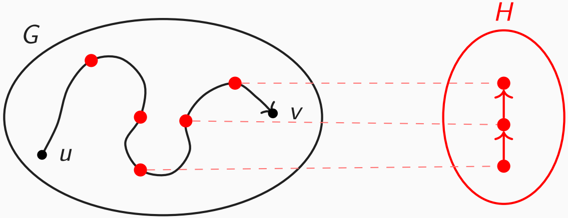


Choose $\lambda := 2^{\Theta(\sqrt{\log n})}$, find a set S of n/λ vertices such that

- s, v connected in $G \Rightarrow \exists\ s, v$-path that hits S every λ steps

Sparsification H on S: $(u, v) \in E(H) \Leftrightarrow \mathrm{dist}_G(u, v) \leq \lambda$

BBRS: Sparsification



Choose $\lambda := 2^{\Theta(\sqrt{\log n})}$, find a set S of n/λ vertices such that

- s, v connected in $G \Rightarrow \exists\ s, v$-path that hits S every λ steps

Sparsification H on S: $(u, v) \in E(H) \Leftrightarrow \mathrm{dist}_G(u, v) \leq \lambda$

The only remaining query:

- Is u, v an edge in H? $\equiv$ Is there a λ-short path from u to v?

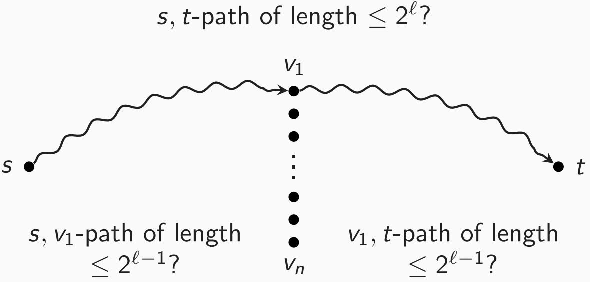
BBRS: Short paths

Is u, v an edge in H? $\equiv$ Is there a λ-short path from u to v?

BBRS: Short paths

Is u, v an edge in H? $\equiv$ Is there a λ-short path from u to v?

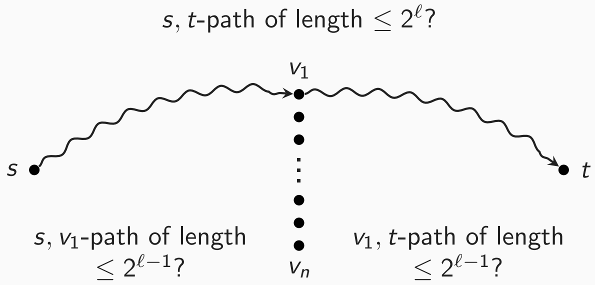
Use **Savitch**:



BBRS: Short paths

Is u, v an edge in H? $\equiv$ Is there a λ-short path from u to v?

Use **Savitch**:



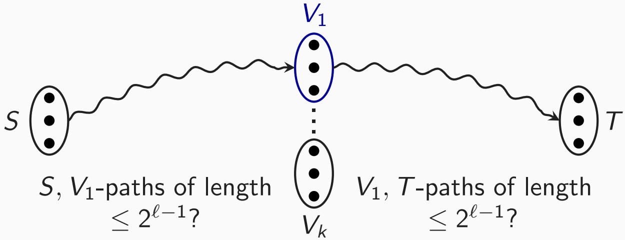
Space: $\mathcal{O}(\log n \cdot \log(\text{distance}))$
Time: $\mathcal{O}(n^{\log(\text{distance})})$

BBRS: Short paths

Is u, v an edge in H? $\equiv$ Is there a λ-short path from u to v?

Use **Savitch**:

S, T-paths of length $\leq 2^{\ell}$?



S, V_1-paths of length $\leq 2^{\ell-1}$?

V_1, T-paths of length $\leq 2^{\ell-1}$?

Batch size $n/\lambda = n/2^{\Theta(\sqrt{\log n})}$

Space: $\mathcal{O}(\log n \cdot \log(\text{distance}))$
Time: $\mathcal{O}(n^{\log(\text{distance})})$

Batched: $n/2^{\Omega(\sqrt{\log n})}$
Batched: $\text{poly}(n)$

Solving STCONN the BBRS way

Two algorithms and a sparsification (with $\lambda = 2^{\Theta(\sqrt{\log n})}$):

Solving STCONN the BBRS way

Two algorithms and a sparsification (with $\lambda = 2^{\Theta(\sqrt{\log n})}$):

- A representative set $S \subseteq V(G)$ of size $n/\lambda \rightsquigarrow$ sparsified graph H on S, $(u, v) \in E(H) \Leftrightarrow \mathrm{dist}_G(u, v) \leq \lambda$

Solving STCONN **the BBRS way**

Two algorithms and a sparsification (with $\lambda = 2^{\Theta(\sqrt{\log n})}$):

- A representative set $S \subseteq V(G)$ of size $n/\lambda \rightsquigarrow$ sparsified graph H on S, $(u, v) \in E(H) \Leftrightarrow \mathrm{dist}_G(u, v) \leq \lambda$
- Long-path algorithm: "Is there a path from u to v in H?"

Solving STCONN the BBRS way

Two algorithms and a sparsification (with $\lambda = 2^{\Theta(\sqrt{\log n})}$):

- A representative set $S \subseteq V(G)$ of size $n/\lambda \rightsquigarrow$ sparsified graph H on S, $(u, v) \in E(H) \Leftrightarrow \mathrm{dist}_G(u, v) \leq \lambda$
- Long-path algorithm: "Is there a path from u to v in H?"
- Short-path algorithm: "Is there a λ-length path from u to v in G?"

Applying BBRS with catalytic space

Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\mathrm{poly}(n)$ time.

Applying BBRS with catalytic space

Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\mathrm{poly}(n)$ time.

1. Build a "simply constructible" representative set U

Applying BBRS with catalytic space

Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and poly(n) time.

1. Build a “simply constructible” representative set U
2. Long walks: catalytic BFS

Applying BBRS with catalytic space

Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and poly(n) time.

1. Build a "simply constructible" representative set U
2. Long walks: catalytic BFS
3. Short walks: catalytic Savitch

Applying BBRS with catalytic space

Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\mathrm{poly}(n)$ time.

1. Build a "simply constructible" representative set U
2. Long walks: catalytic BFS – Cook, Pyne'26
3. Short walks: catalytic Savitch

Applying BBRS with catalytic space

Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\mathrm{poly}(n)$ time.

1. **Build a "simply constructible" representative set U**
2. Long walks: catalytic BFS – Cook, Pyne'26
3. **Short walks: catalytic Savitch**

Applying BBRS with catalytic space

Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\mathrm{poly}(n)$ time.

1. **Build a "simply constructible" representative set** U
2. Long walks: catalytic BFS – Cook, Pyne'26
3. **Short walks: catalytic Savitch** – concurrently Edenhofer'26

Applying BBRS with catalytic space

Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\mathrm{poly}(n)$ time.

1. **Build a "simply constructible" representative set** U
2. Long walks: catalytic BFS – Cook, Pyne'26
3. **Short walks: catalytic Savitch** – concurrently Edenhofer'26

Finding the set U

- BBRS use averaging argument

Applying BBRS with catalytic space

Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and poly(n) time.

1. **Build a "simply constructible" representative set** U
2. Long walks: catalytic BFS – Cook, Pyne'26
3. **Short walks: catalytic Savitch** – concurrently Edenhofer'26

Finding the set U

- BBRS use averaging argument
- We use sampling via pairwise independence and random walks on expanders

Applying BBRS with catalytic space

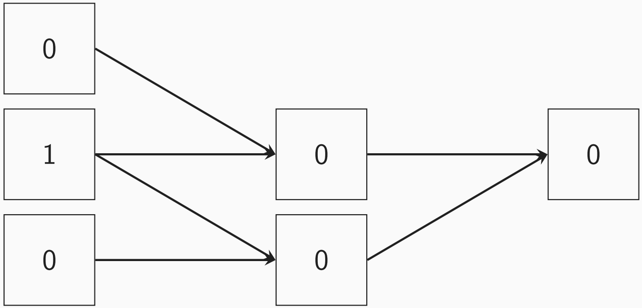
Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and poly(n) time.

1. **Build a "simply constructible" representative set** U ✓
2. Long walks: catalytic BFS – Cook, Pyne'26
3. **Short walks: catalytic Savitch** – concurrently Edenhofer'26

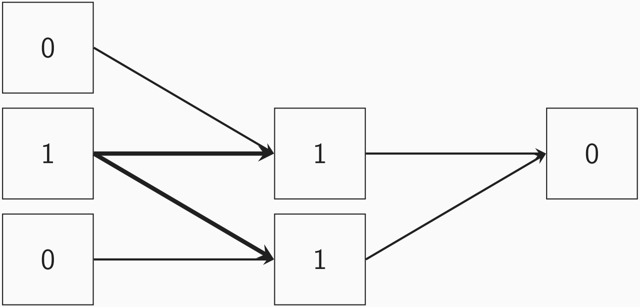
Finding the set U

- BBRS use averaging argument
- We use sampling via pairwise independence and random walks on expanders

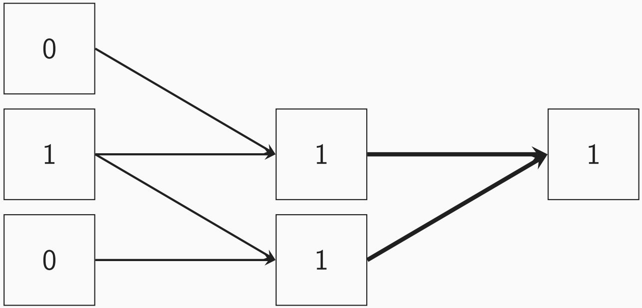
What is BFS?



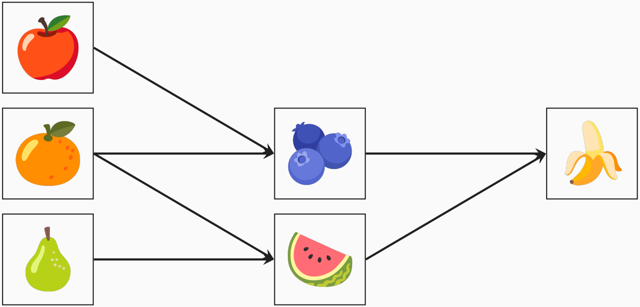
What is BFS?



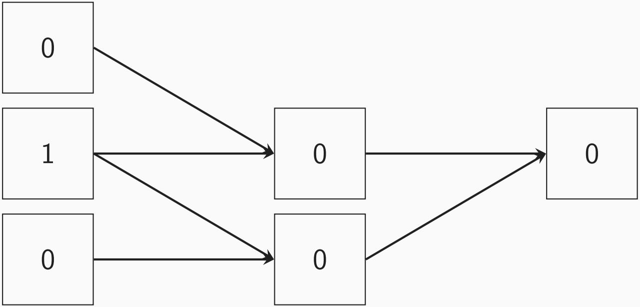
What is BFS?



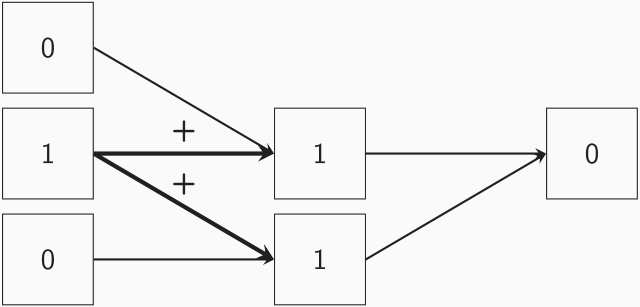
Oh no



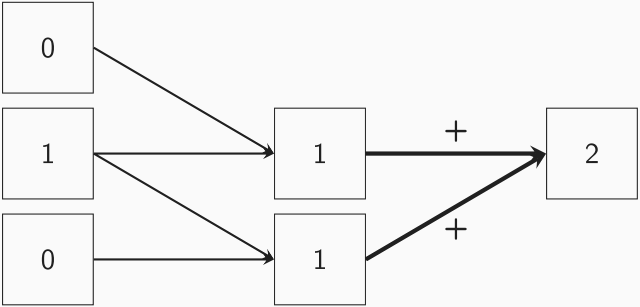
What if we use arithmetic?



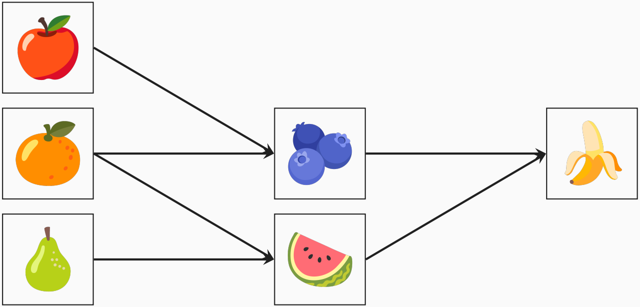
What if we use arithmetic?



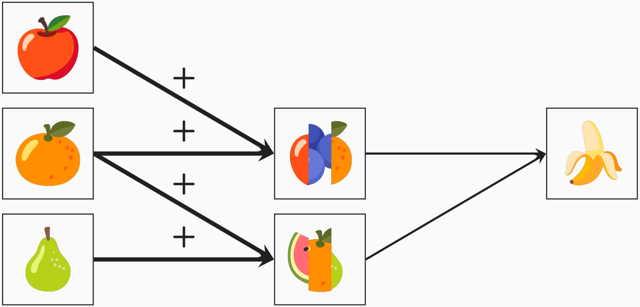
What if we use arithmetic?



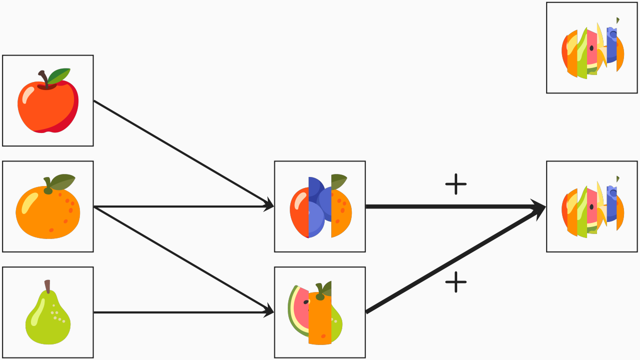
What do Cook and Pyne do?



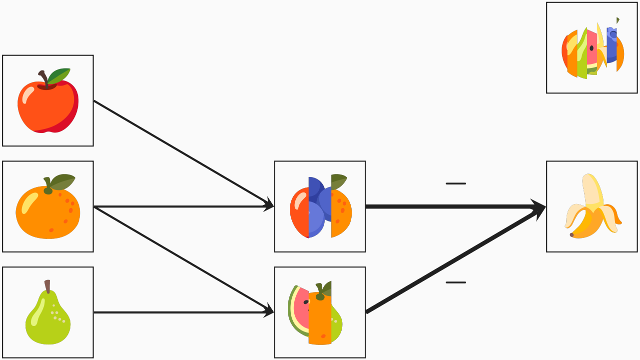
What do Cook and Pyne do?



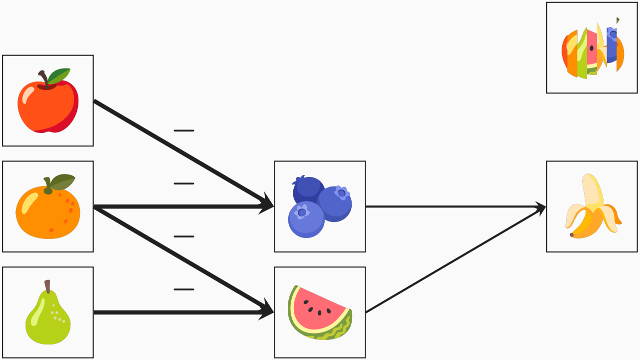
What do Cook and Pyne do?



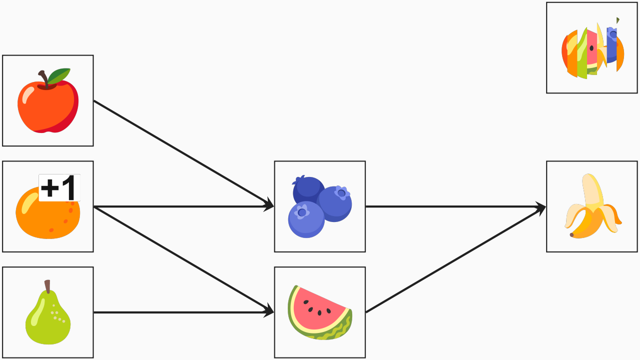
What do Cook and Pyne do?



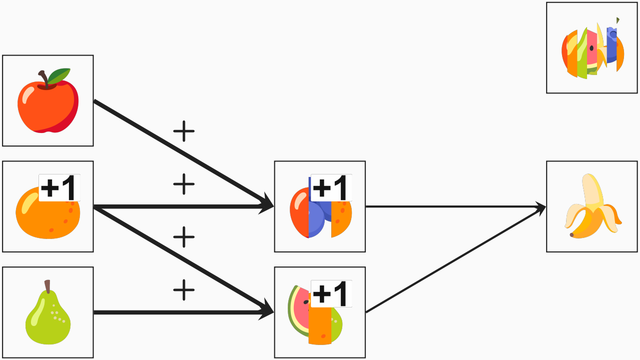
What do Cook and Pyne do?



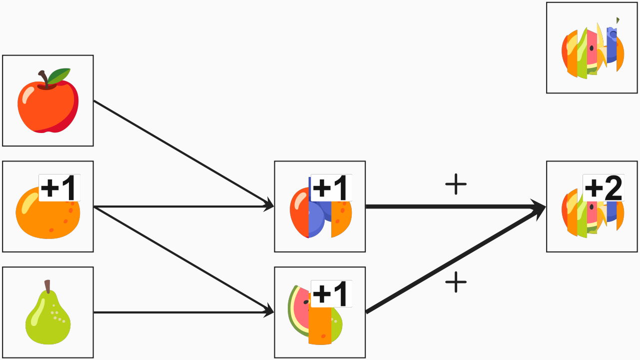
What do Cook and Pyne do?



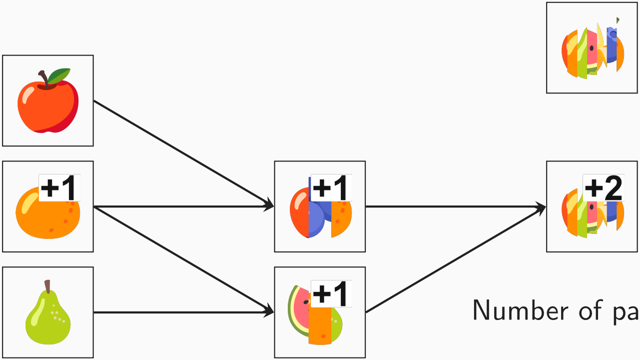
What do Cook and Pyne do?



What do Cook and Pyne do?

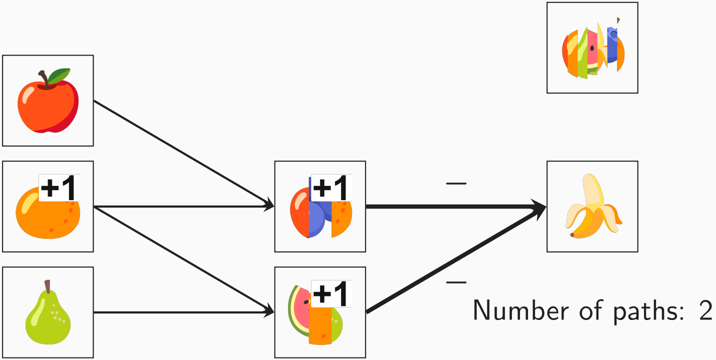


What do Cook and Pyne do?

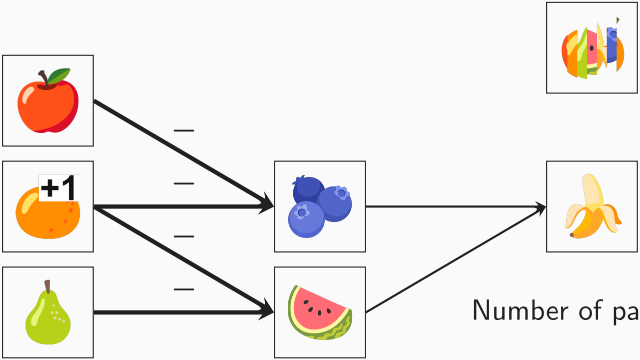


Number of paths: 2

What do Cook and Pyne do?

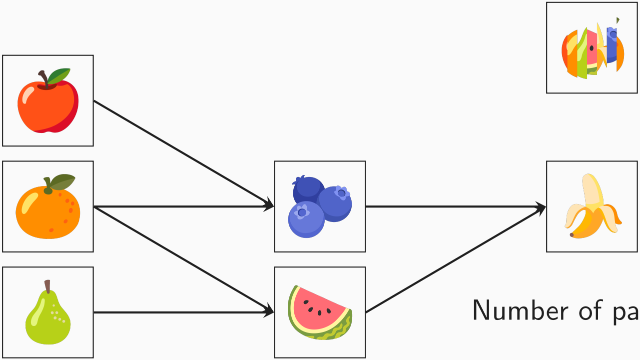


What do Cook and Pyne do?



Number of paths: 2

What do Cook and Pyne do?



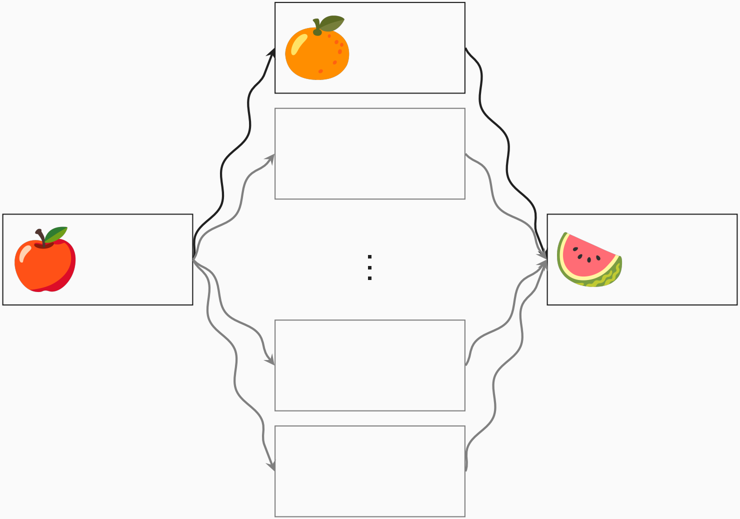
Number of paths: 2

Applying BBRS with catalytic space

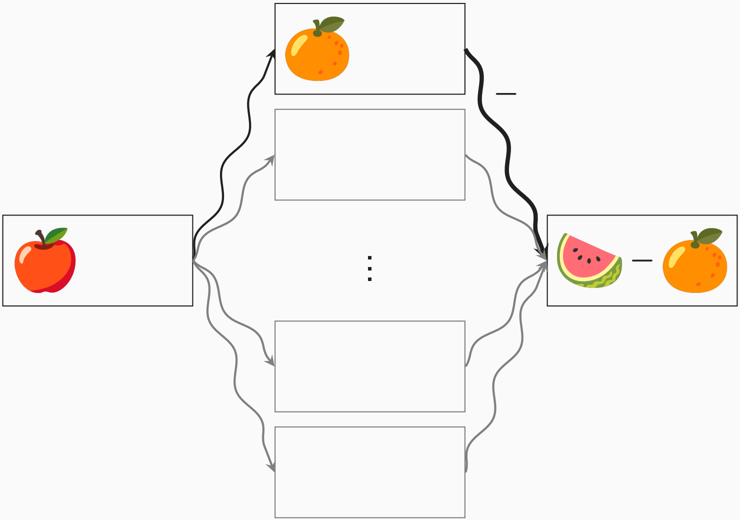
Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and poly(n) time.

1. **Build a "simply constructible" representative set** U ✓
2. Long walks: catalytic BFS – Cook, Pyne'26 ✓
3. **Short walks: catalytic Savitch** – concurrently Edenhofer'26

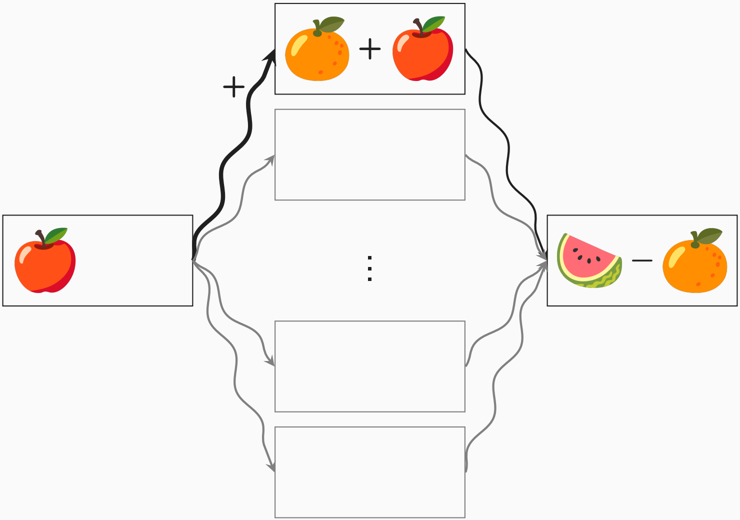
Catalytic Savitch



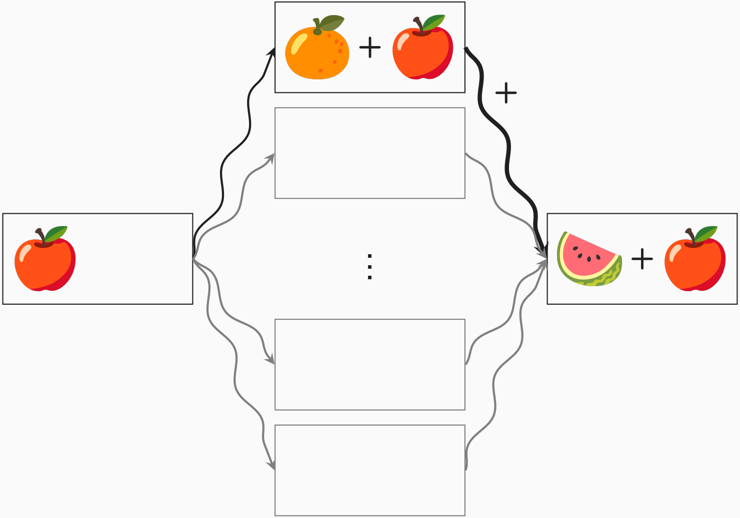
Catalytic Savitch



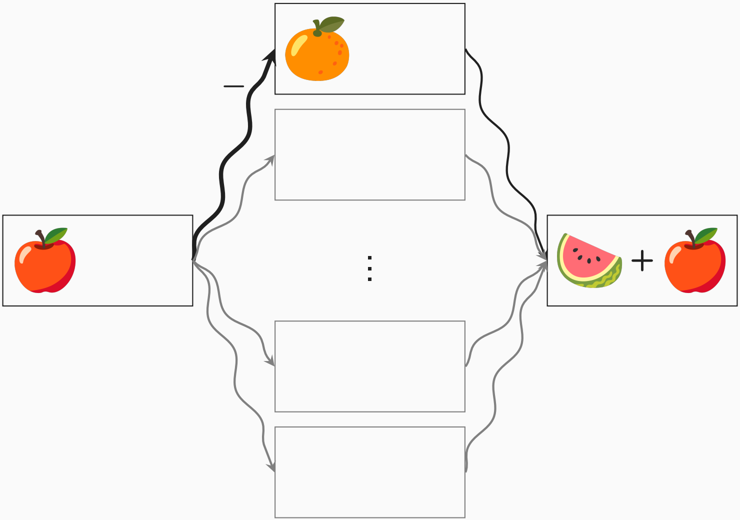
Catalytic Savitch



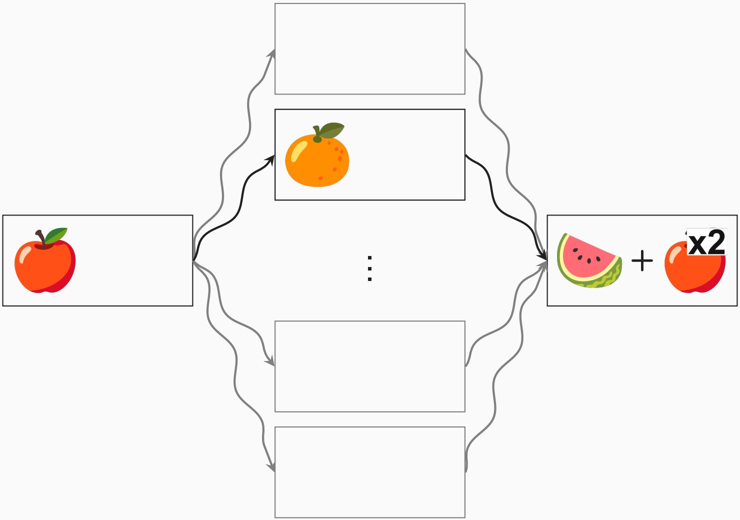
Catalytic Savitch



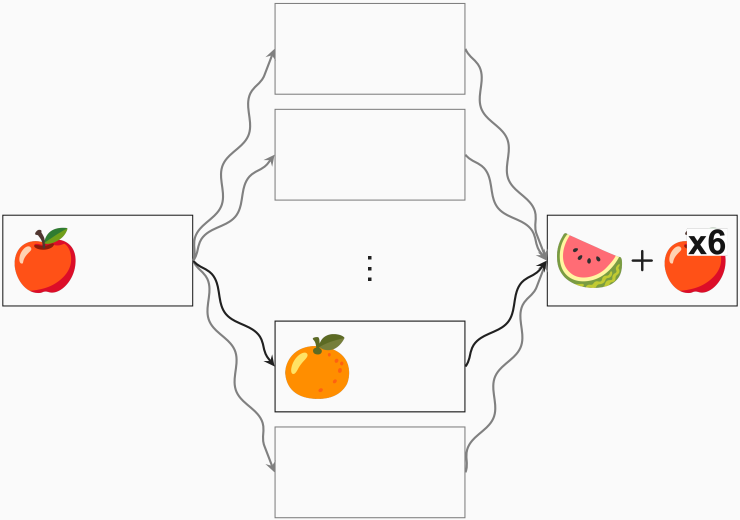
Catalytic Savitch



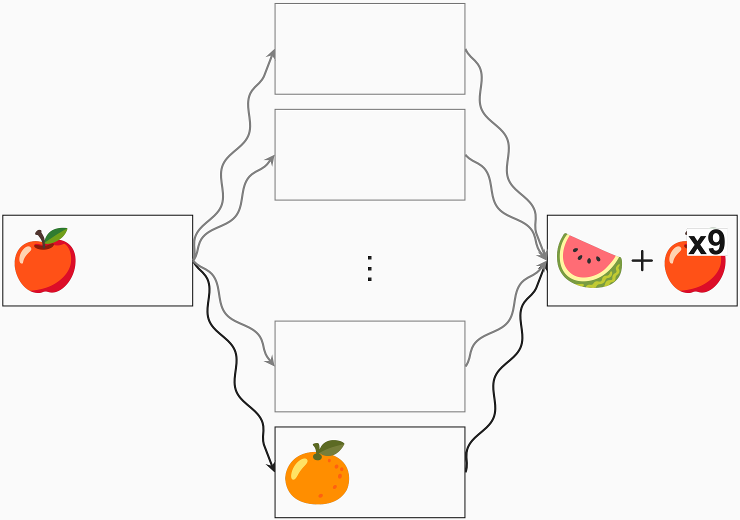
Catalytic Savitch



Catalytic Savitch



Catalytic Savitch



Applying BBRS with catalytic space

Reminder of Theorem 1
STCONN can be solved in $\mathcal{O}(\log n)$ free space, $n/2^{\Omega(\sqrt{\log n})}$ catalytic space and $\mathrm{poly}(n)$ time.

1. **Build a "simply constructible" representative set** U ✓
2. Long walks: catalytic BFS – Cook, Pyne'26 ✓
3. **Short walks: catalytic Savitch** – concurr. Edenhofer'26 ✓

Open questions

- A poly-time algorithm using space smaller than $n/2^{\Theta(\sqrt{\log n})}$?

Open questions

- A poly-time algorithm using space smaller than $n/2^{\Theta(\sqrt{\log n})}$?
- An interesting double-sided error algorithm for STCONN?

Open questions

- A poly-time algorithm using space smaller than $n/2^{\Theta(\sqrt{\log n})}$?
- An interesting double-sided error algorithm for STCONN?

Thank you!