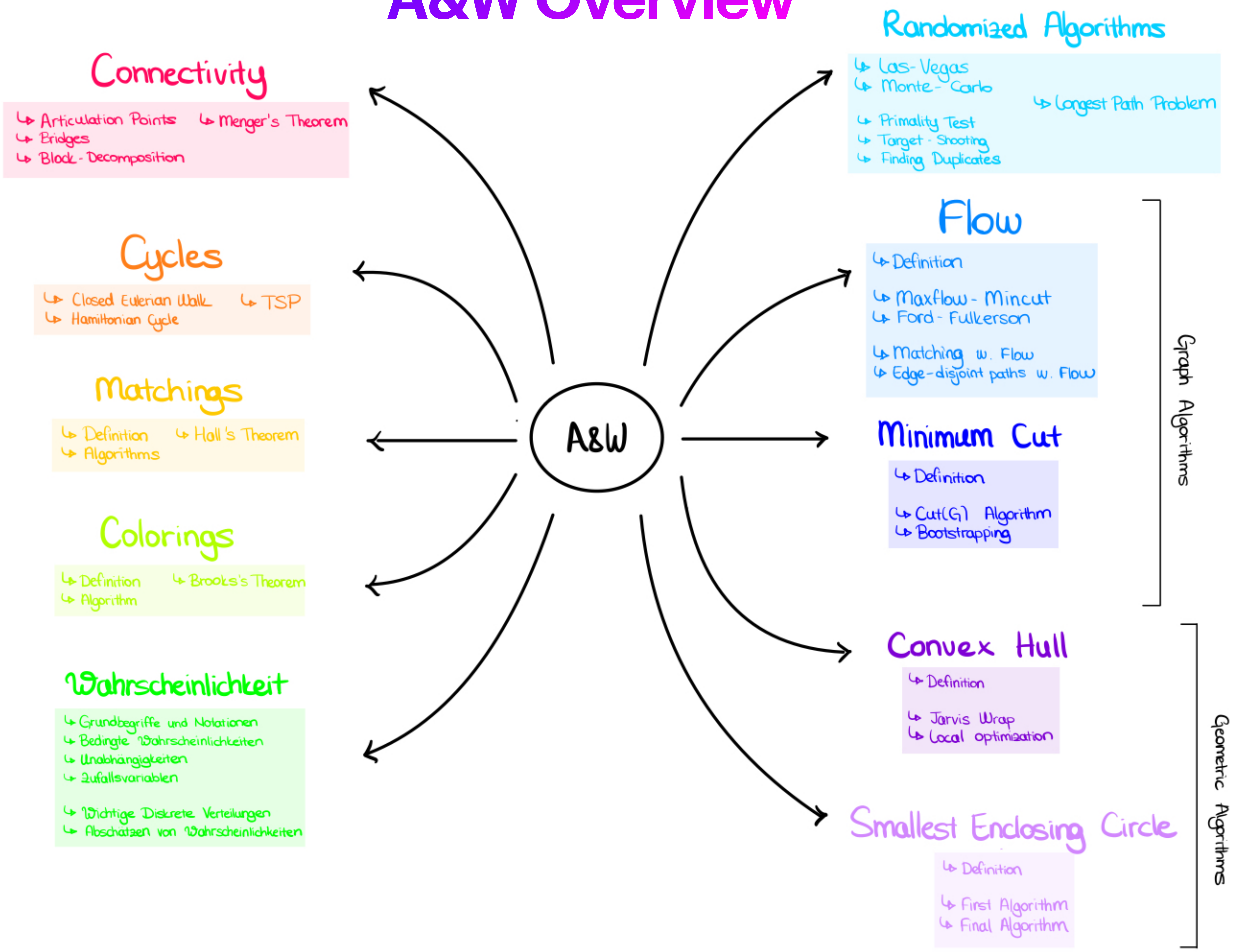
A&W

Exercise Session 3

Cycles , TSP

Nil Ozer

A&W Overview



A&W
Connectivity
↳ Articulation Points
↳ Bridges
↳ Block-Decomposition
↳ Menger's Theorem
Cycles
↳ Closed Eulerian Walk
↳ Hamiltonian Cycle
↳ TSP
Matchings
↳ Definition
↳ Algorithms
↳ Hall's Theorem
Colorings
↳ Definition
↳ Algorithm
↳ Brooks's Theorem
Wahrscheinlichkeit
↳ Grundbegriffe und Notationen
↳ Bedingte Wahrscheinlichkeiten
↳ Unabhängigkeiten
↳ Zufallsvariablen
↳ Wichtige Diskrete Verteilungen
↳ Abschätzen von Wahrscheinlichkeiten
Randomized Algorithms
↳ Las-Vegas
↳ Monte-Carlo
↳ Primality Test
↳ Target-Shooting
↳ Finding Duplicates
↳ Longest Path Problem
Flow
↳ Definition
↳ Maxflow-Mincut
↳ Ford-Fulkerson
↳ Matching w. Flow
↳ Edge-disjoint paths w. Flow
Minimum Cut
↳ Definition
↳ Cut(G) Algorithm
↳ Bootstrapping
Graph Algorithms
Convex Hull
↳ Definition
↳ Jarvis Wrap
↳ Local optimization
Smallest Enclosing Circle
↳ Definition
↳ First Algorithm
↳ Final Algorithm
Geometric Algorithms

Outline

- Some logistics
- Connectivity Kahoot
- Cycles
- TSP

Logistics

- For every exercise, you'll receive **feedback** from me on the exercise session next week !

- **Anki** cards
- **CodeExpert** videos

- Regular recap **kahoots** in class on the weeks without the minitest

Connectivity Kahoot

Cycles

Cycles
Definitions

Closed walk

- A sequence of vertices $(v_0, v_1, \ldots, v_k)$ is a **closed walk** (german "Zyklus") if it is a walk, $k \geq 2$ and $v_0 = v_k$.

Cycle

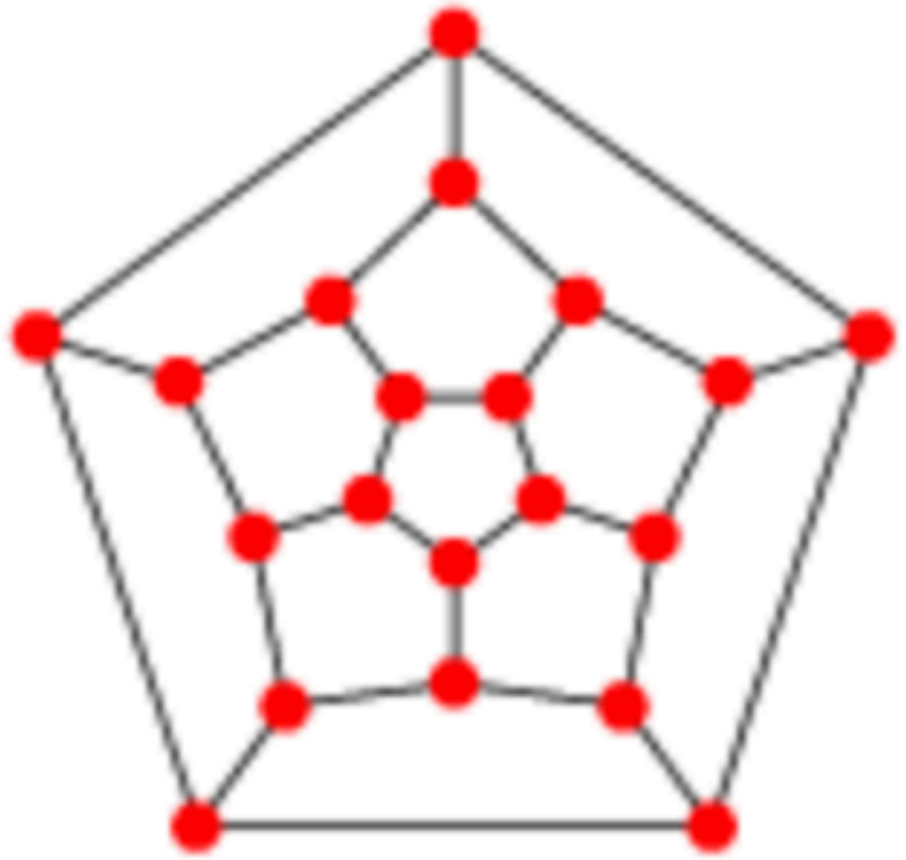
- A sequence of vertices $(v_0, v_1, \ldots, v_k)$ is a **cycle** (german "Kreis") if it is a closed walk, $k \geq 3$ and all vertices (except v_0 and v_k) are distinct.

- Hamlitonian Cycle
  - A cycle in G that contains every vertex exactly once

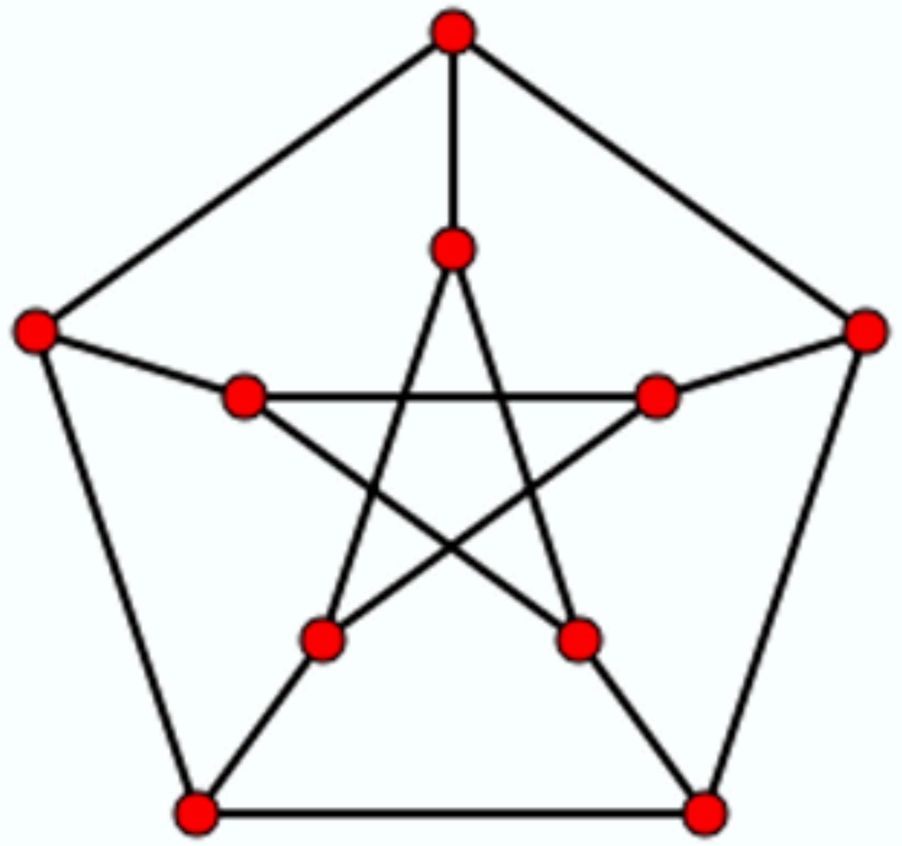
- Eulerian Cycle
  - A closed walk in G that contains every edge exactly once

Cycles
Hamiltonian Cycle Examples

- Hamlitonian Cycle
  - A cycle in G that contains every vertex exactly once



Ikosaeder

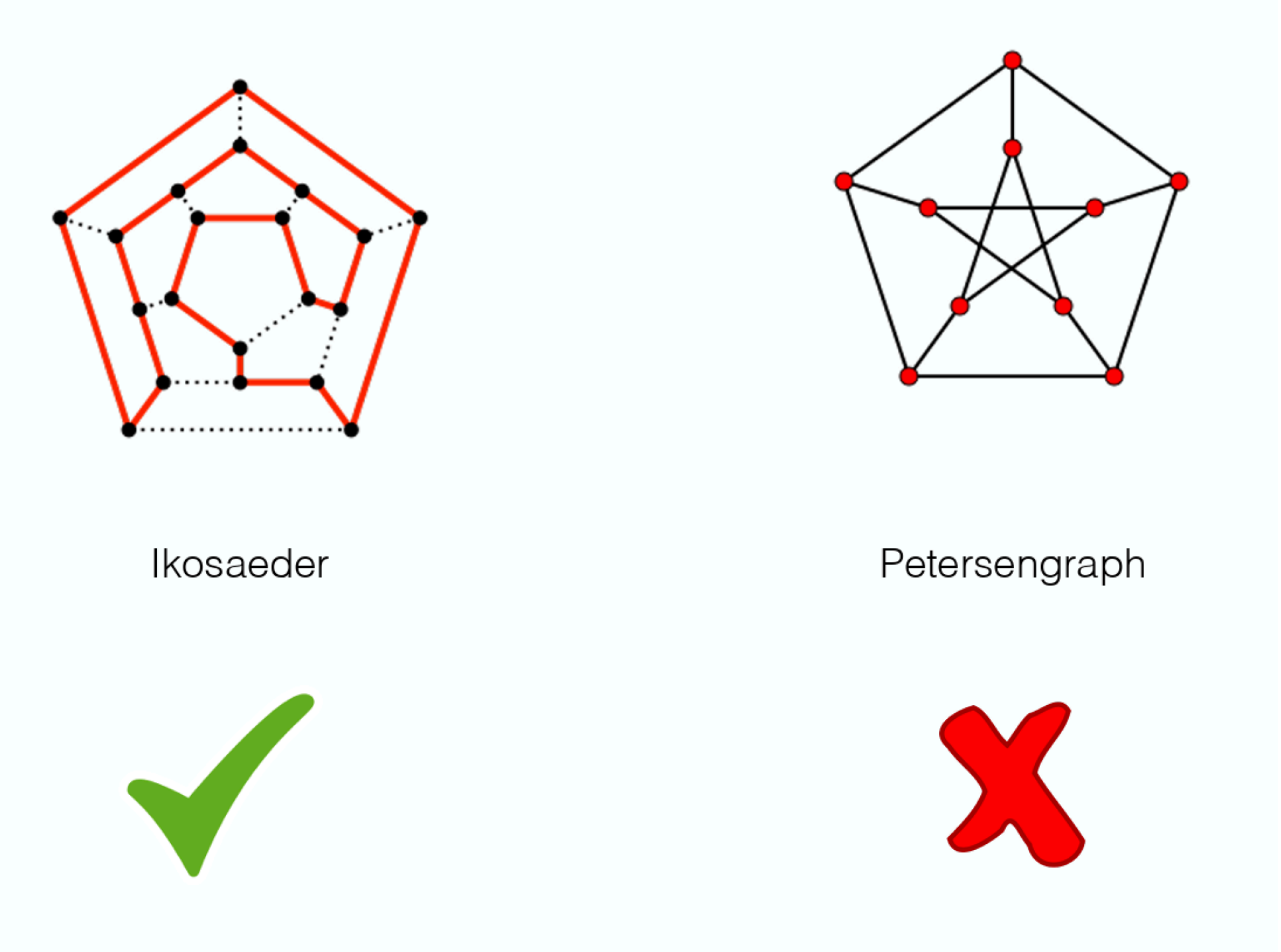


Petersengraph

Cycles

Hamiltonian Cycle Examples

- Hamlitonian Cycle
  - A cycle in G that contains every vertex exactly once

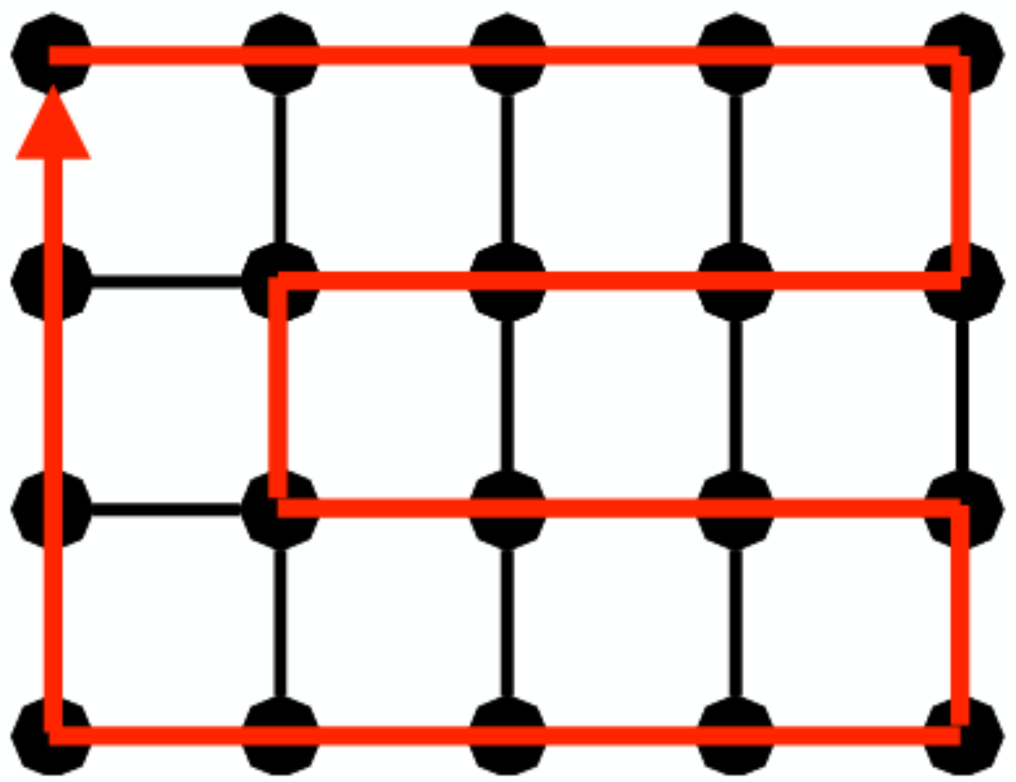


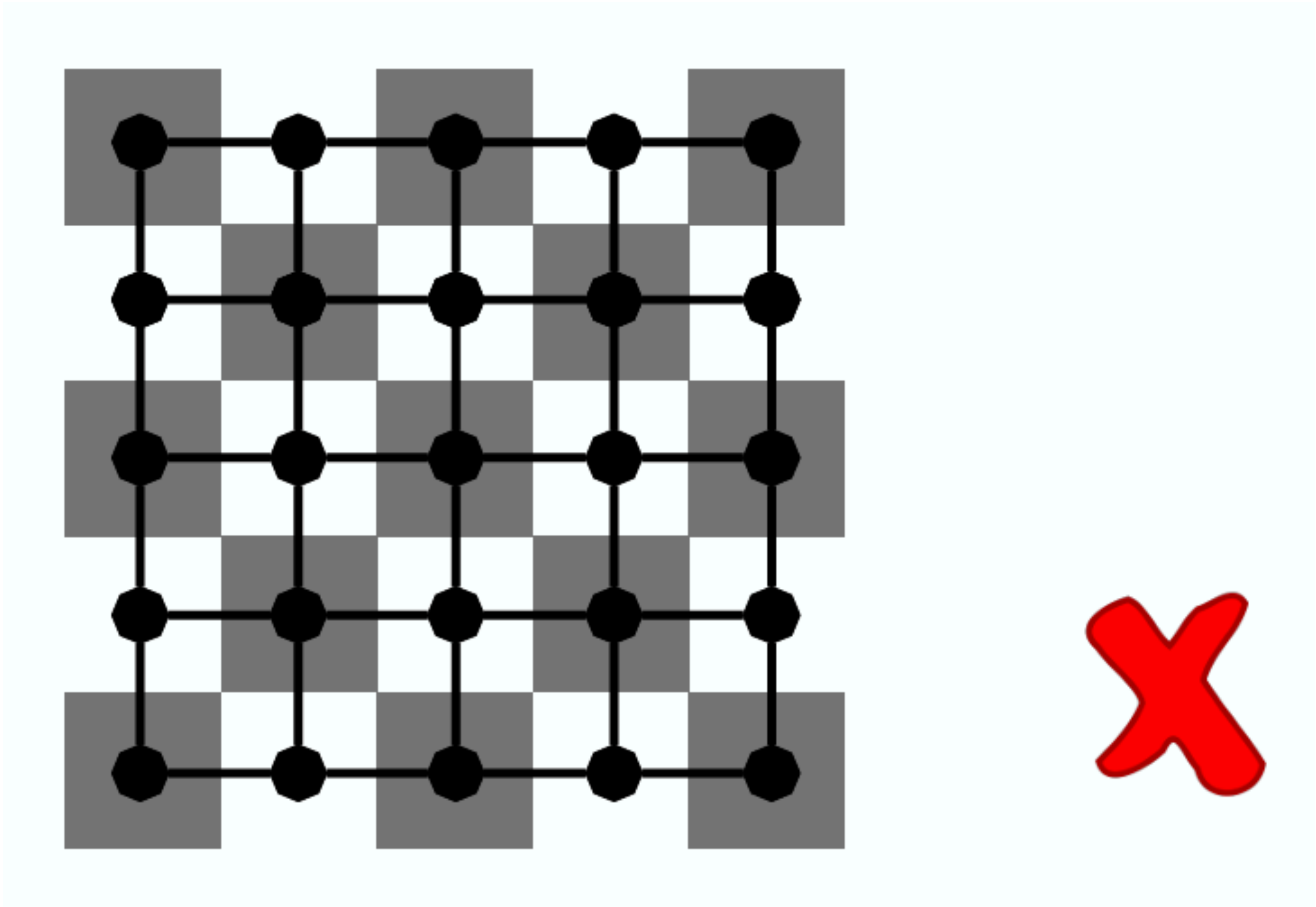
Cycles

Hamiltonian Cycle Examples

- Hamlitonian Cycle
  - A cycle in G that contains every vertex exactly once

Grid Graph





Let $m, n \geq 2$

A n x m Grid has a hamiltonian cycle iff n x m is even

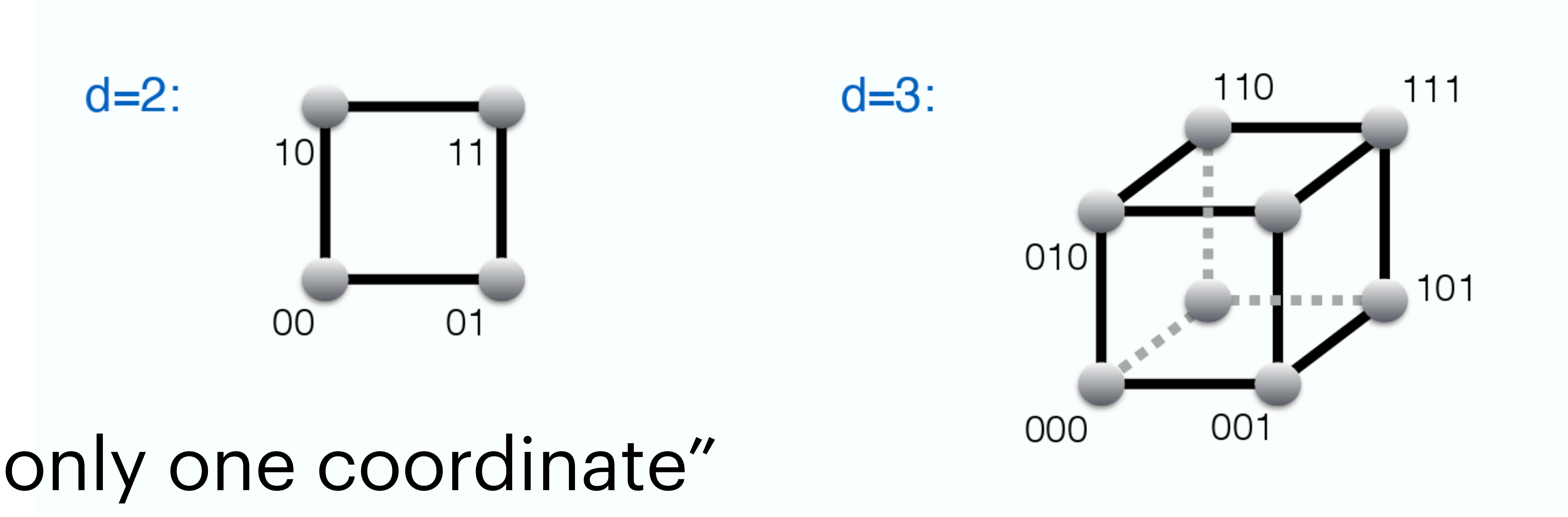
Cycles
Hamiltonian Cycle Examples

- Hamlitonian Cycle
  - A cycle in G that contains every vertex exactly once

d-dimensional Hypercube H_d

V := $\{0,1\}^d$

E := "All vertex pairs that differ in only one coordinate"



Has a hamiltonian cycle for all $d \geq 2$

Eulerian Cycle

Lemma

A connected G has a eulerian Cycle $\iff$ Every vertex has an even degree

Let's take a break

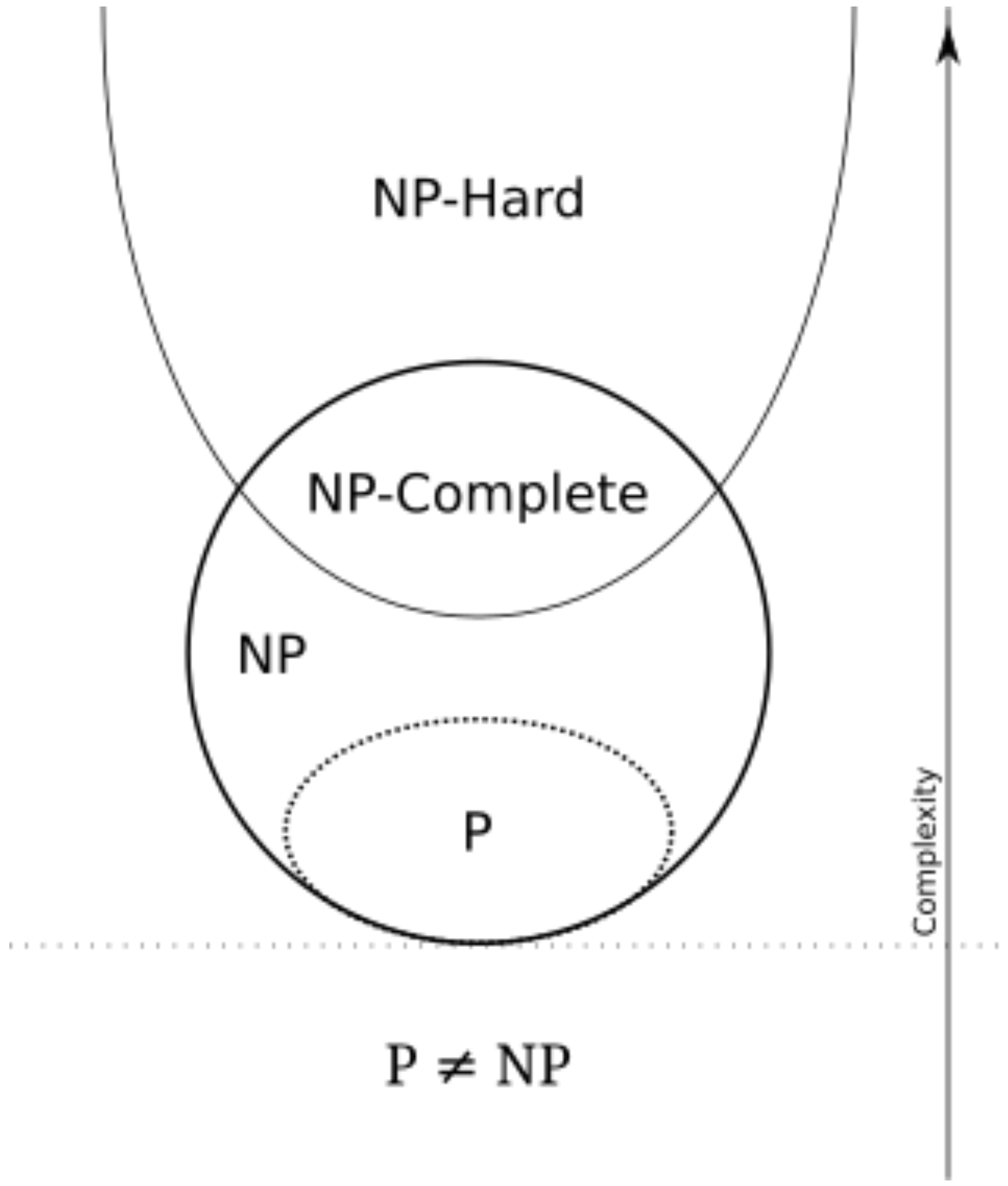
Hamiltonian Cycle

Given a Graph $G = (V, E)$, does G have a hamiltonian cycle ?

NP - Complete

NP-Complete

Given a Graph $G = (V, E)$, does G have a hamiltonian cycle ?



Complexity Theory

TI next semester

P : polynomial

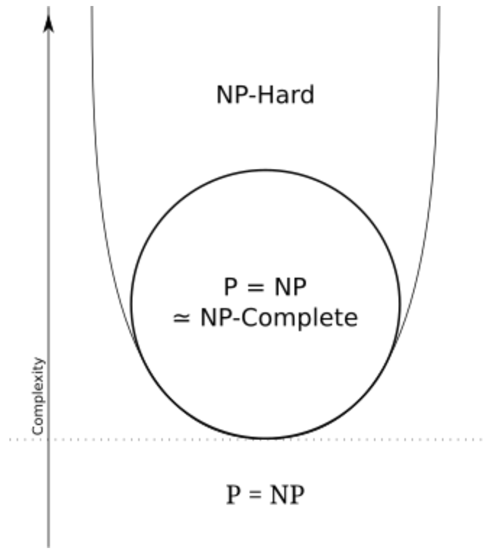
NP : non-deterministic polynomial

- NP is the set of decision problems *solvable* in polynomial time by a nondeterministic Turing machine.
- NP is the set of decision problems *verifiable* in polynomial time by a deterministic Turing machine.

NP - Complete

A problem a in NP is NP-complete if :

$$a \in P \implies P = NP$$



?

Hamiltonian Cycle

Dirac's Theorem

$|V| \geq 3$ and

the minimum degree $\delta(G) \geq |V|/2$ $\Longrightarrow$ A G has a hamiltonian cycle

Hamiltonian Cycle

DP Approach

For all $S \subseteq [n]$ with $1 \in S$ and all $x \in S$ with $x \neq 1$:

$$P[S][x] = \begin{cases} 1 & \text{, if there exists a 1-x-path that only uses vertices from S} \\ 0 & \text{, else} \end{cases}$$

Initialization : P[{1,x}][x] = 1 iff { 1,x } ∈ E

Schleife:

for all s = 3 to n
  for all S ⊆[n] mit 1∈ S und |S| = s:
    for all x ∈ S mit x ≠ 1:

Rekursion

$P_{S,x} = \max \{ P_{S\setminus\{x\},y} : y \in N(x) \cap S, y \neq 1 \}$

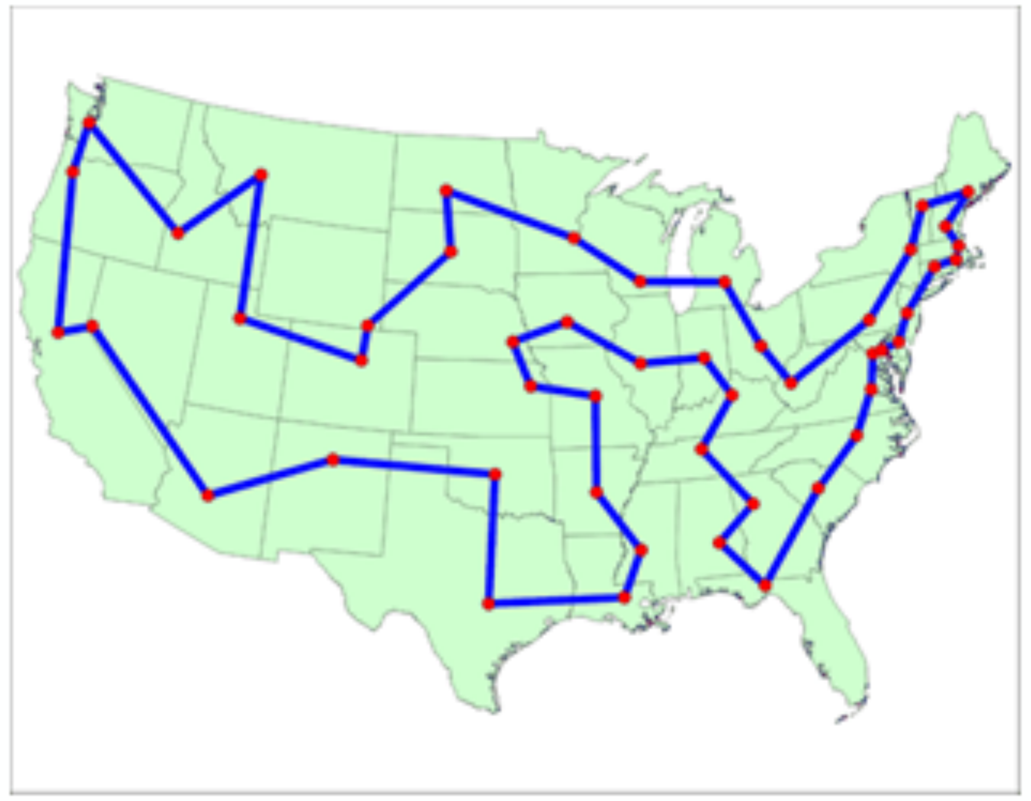
Ausgabe: G enthält Hamiltonkreis gdw $\exists\, x \in N(1)$ mit $P_{[n],x} = 1$

Dynamische Programmierung:
Laufzeit: $\approx n^2\, 2^n$
Speicherplatz: $\approx n\, 2^n$

TSP

TSP

Problem Description



Given : • A complete Graph K_n of n vertices

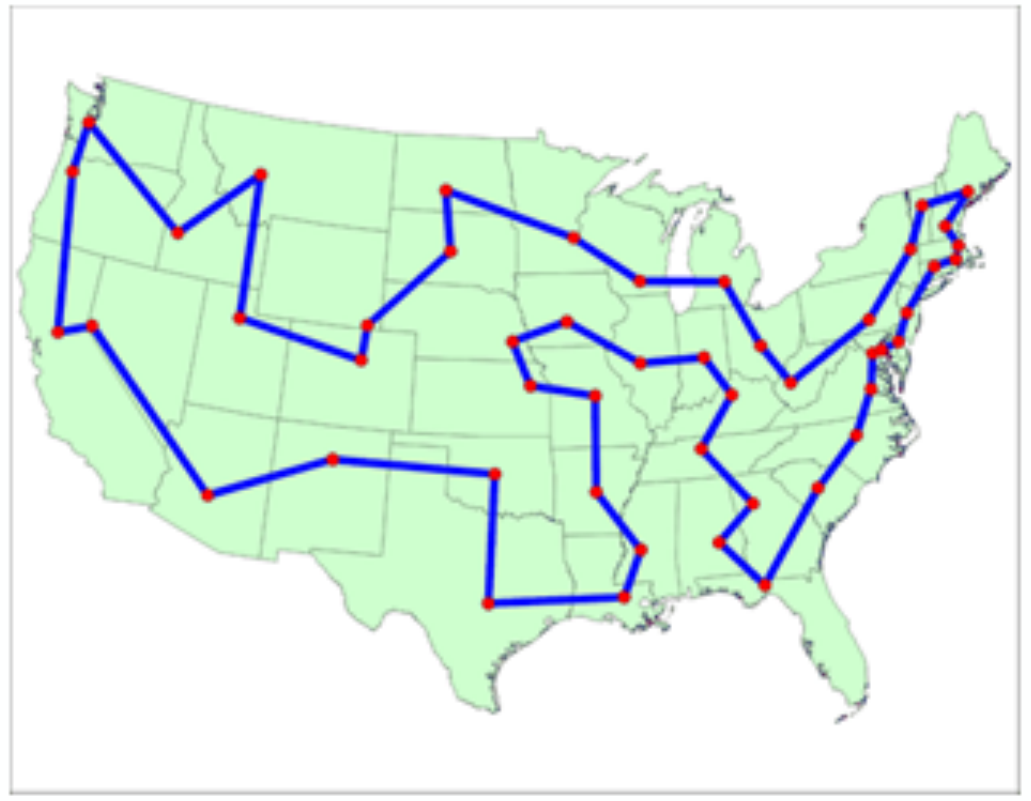
• Distances l inbetween every 2 vertex $l : \binom{[n]}{2} \to R$

To find : • "shortest round trip"

$$\min_{H\,:\,Hamiltonian\ Cycle} \sum_{e \in E(H)} l(e)$$

also NP-Complete

Metric TSP
Problem Description



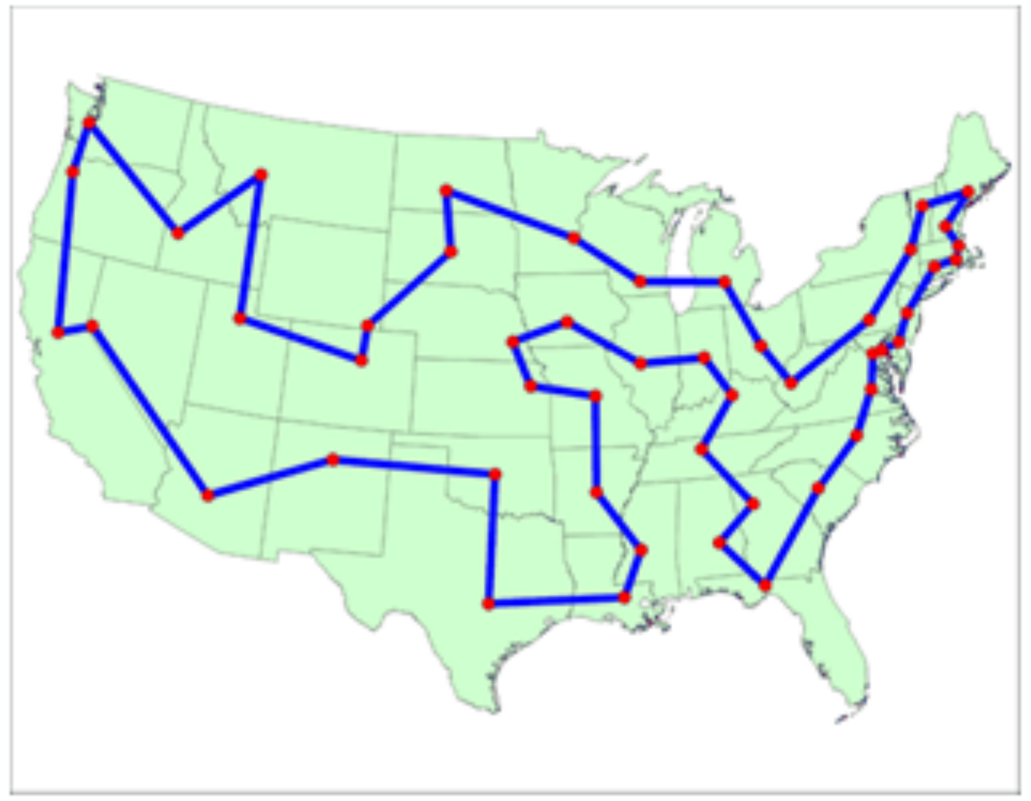
Given :
- A complete Graph K_n of n vertices
- Distances l inbetween every 2 vertex $l : \binom{[n]}{2} \to R$
- l satisfies the triangle inequality $l(x,z) \leq l(x,y) + l(y,z)$

To find :
- "shortest round trip"

$$\min_{H\,:\,Hamiltonian\ Cycle} \sum_{e \in E(H)} l(e)$$

Metric TSP : 2-Approximation

Problem Description



Given :

- A complete Graph K_n of n vertices
- Distances l inbetween every 2 vertex $l : \binom{[n]}{2} \to R$

- l satisfies the triangle inequality $l(x,z) \le l(x,y) + l(y,z)$

To find :

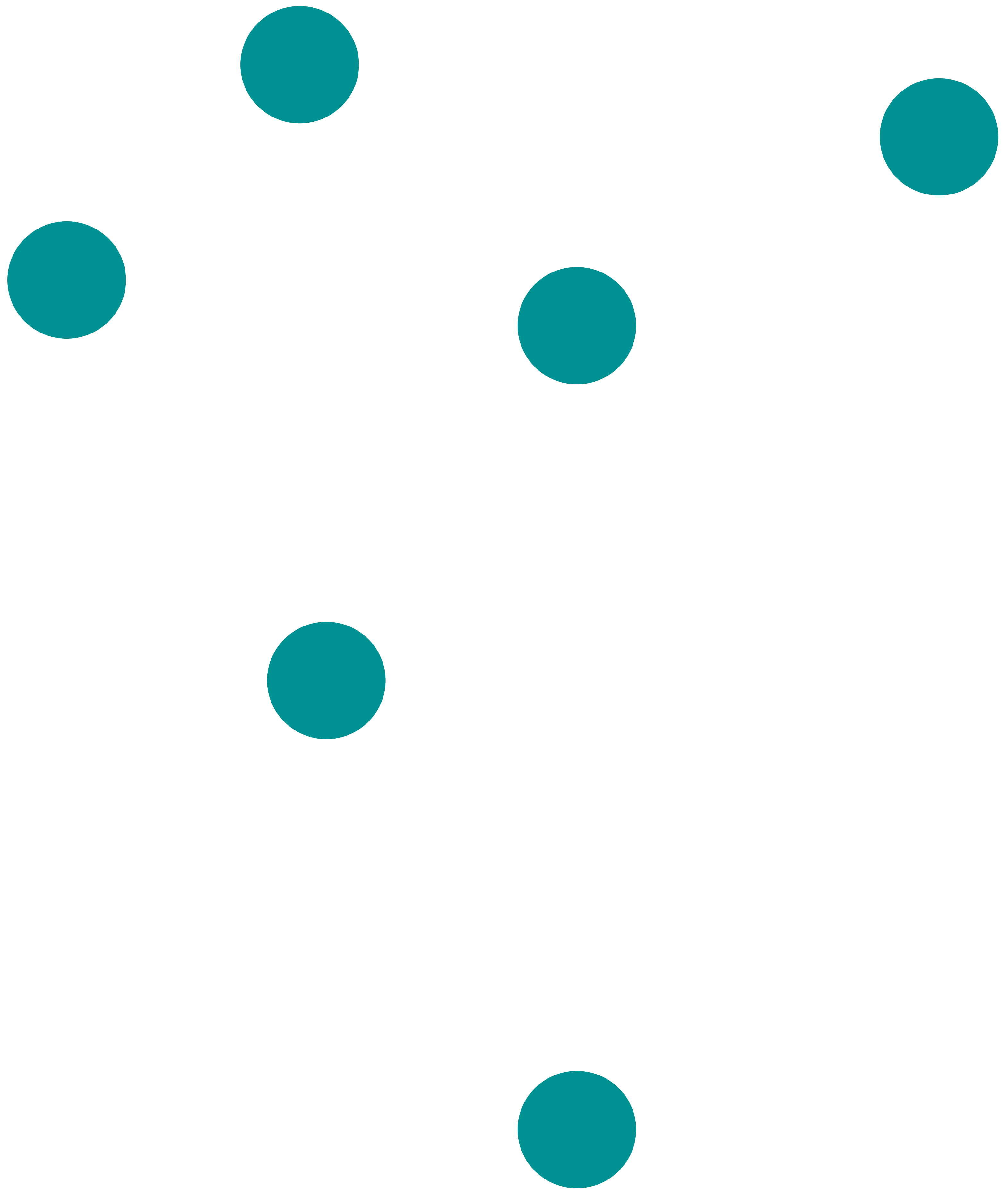
- Hamiltonian Cycle C s.t.

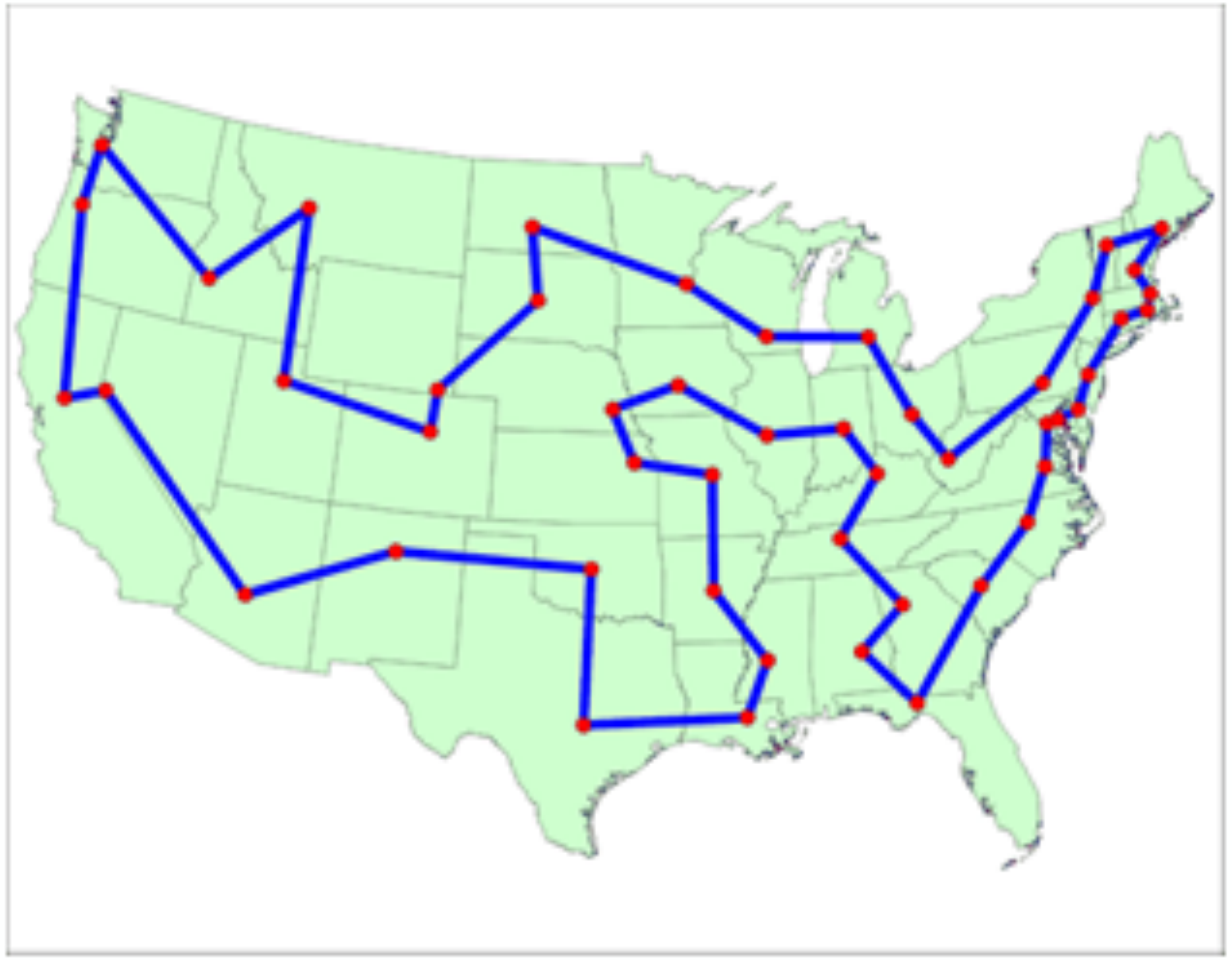
$$l(C) \le 2\, l(OPT)$$

where $OPT = \min_{H\,:\,Hamiltonian\ Cycle} \sum_{e \in E(H)} l(e)$

Metric TSP : 2-Approximation

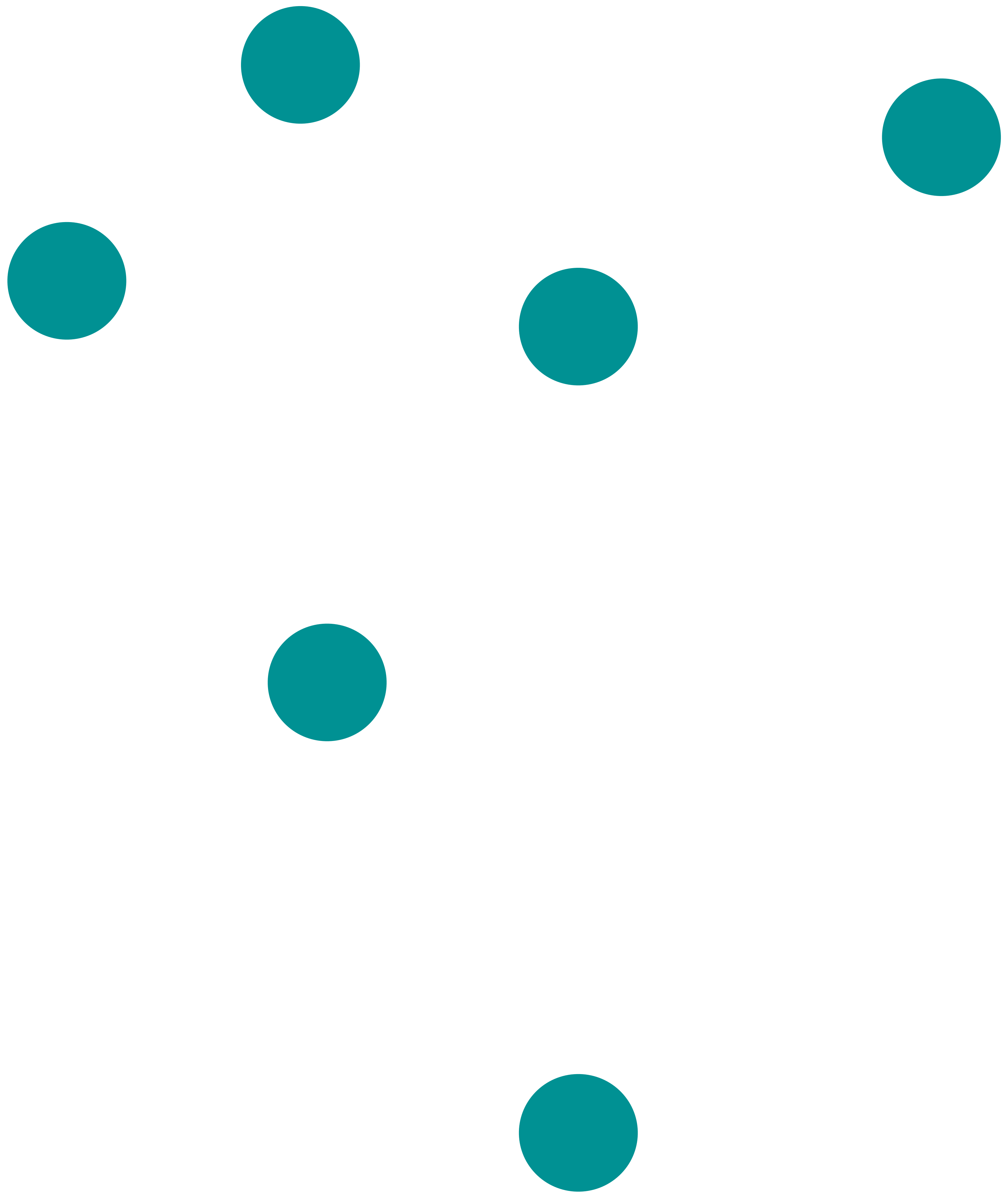
Algorithm





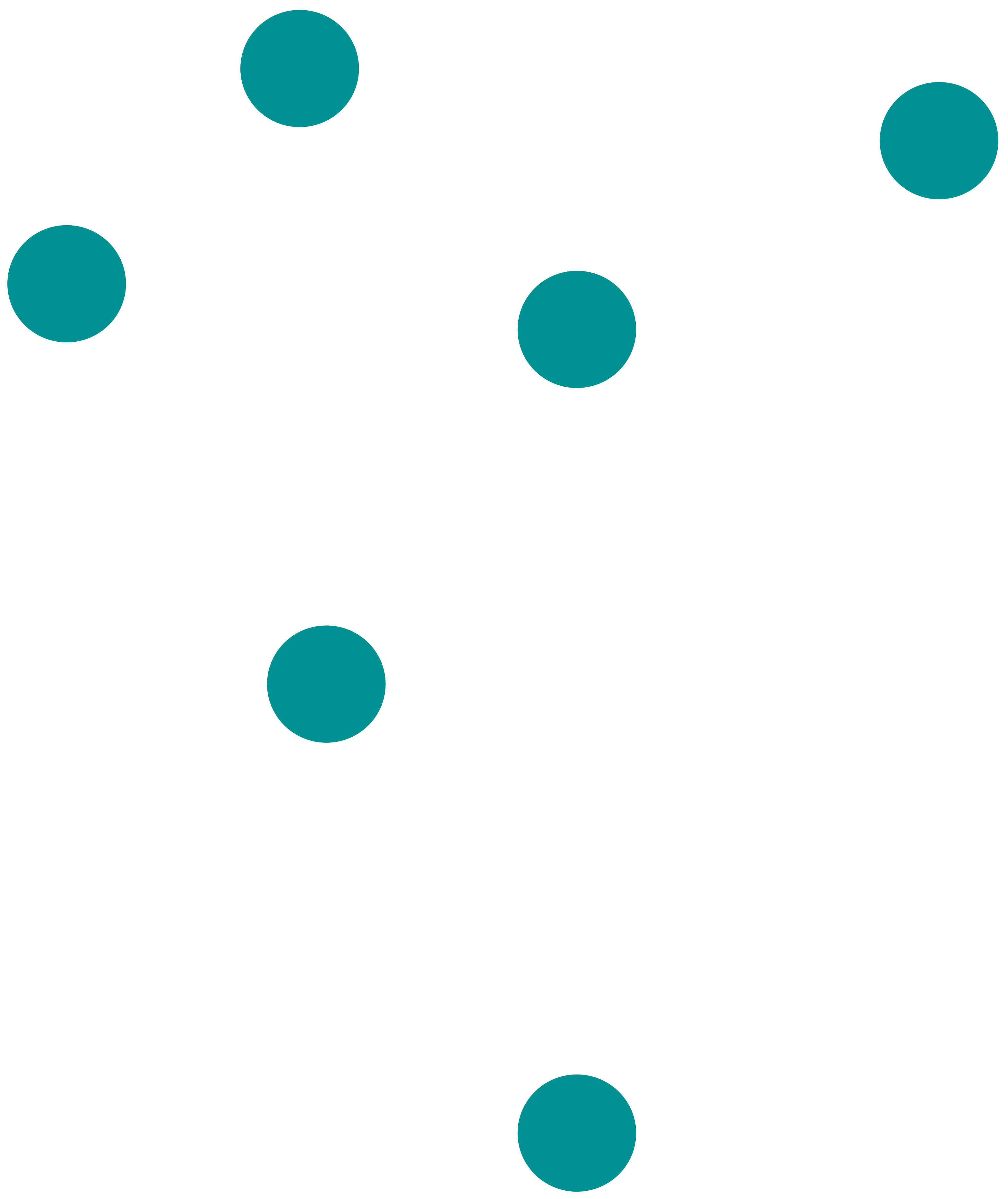
Metric TSP : 2-Approximation

Algorithm



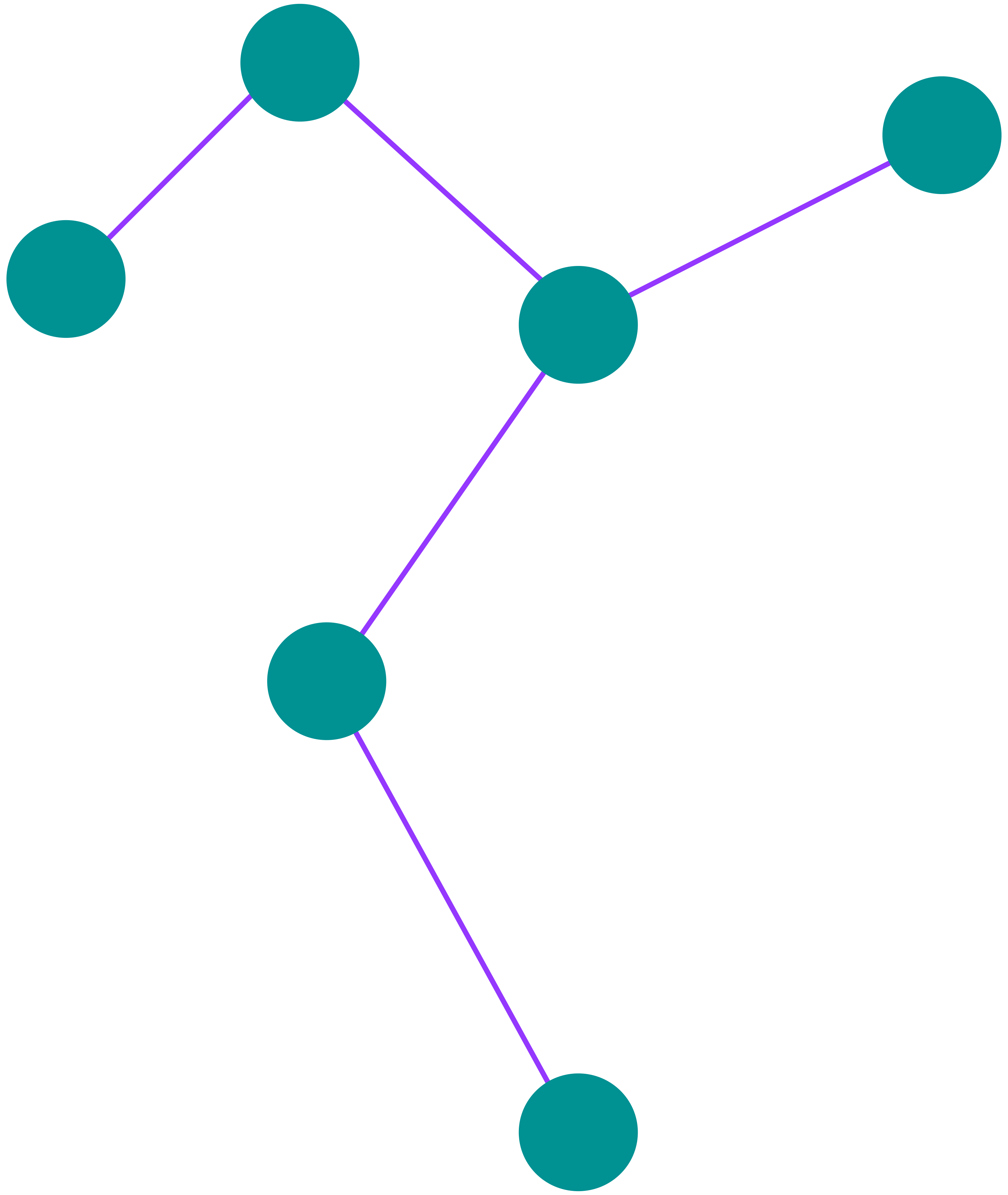
Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

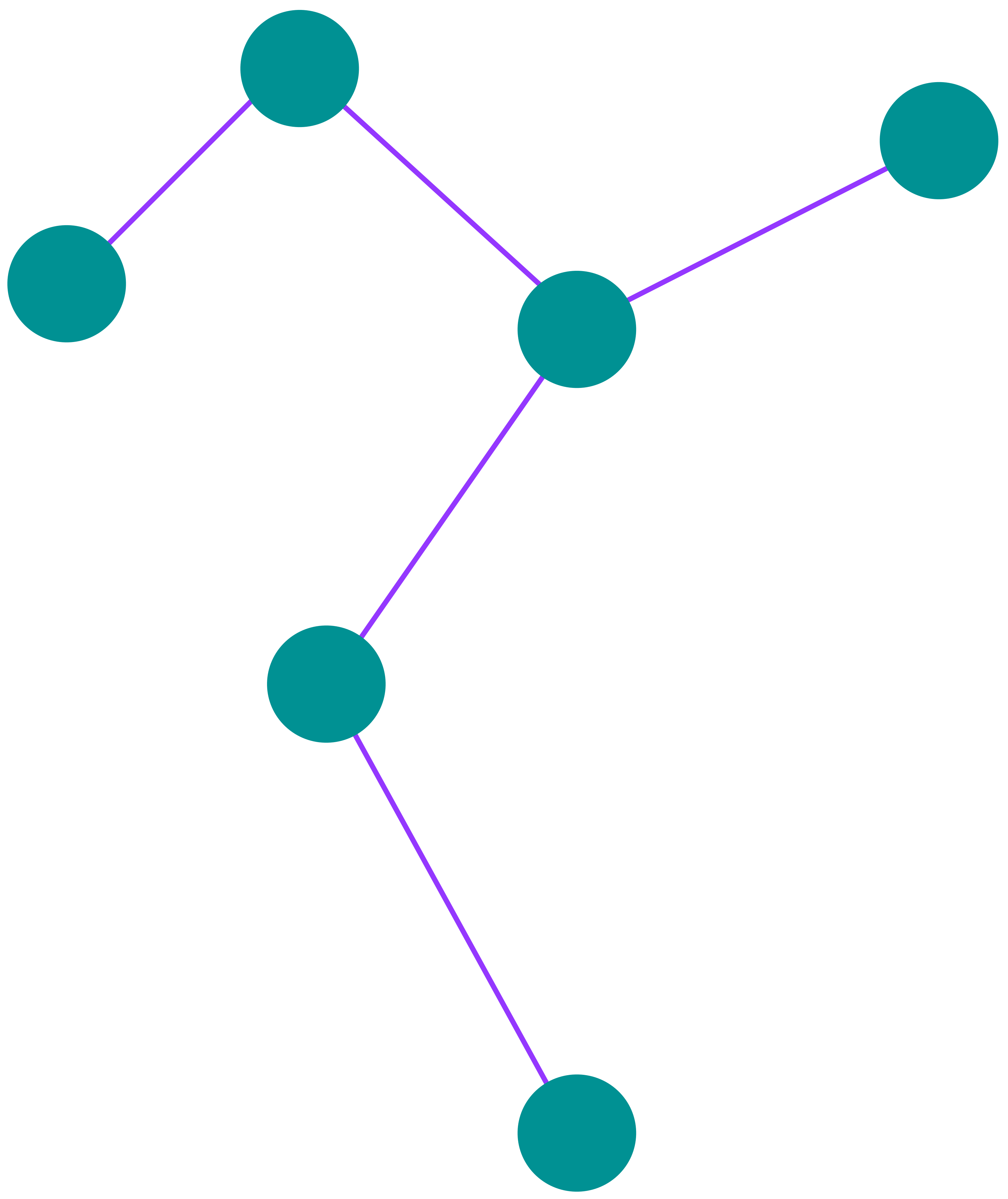
Metric TSP : 2-Approximation
Algorithm



1. Find the MST T

Metric TSP : 2-Approximation

Algorithm

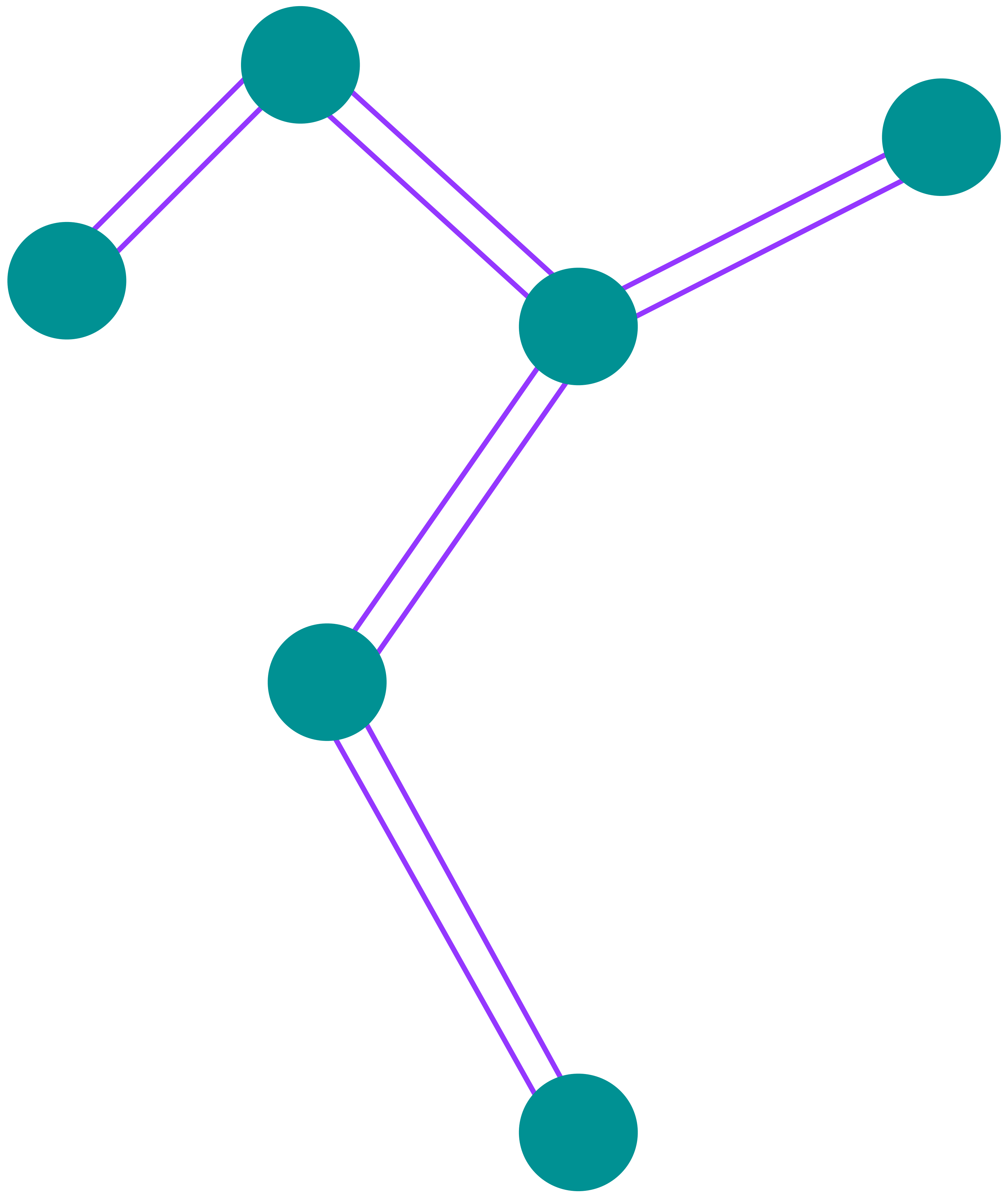


1. Find the MST T

2. Duplicate all edges of T

Metric TSP : 2-Approximation

Algorithm

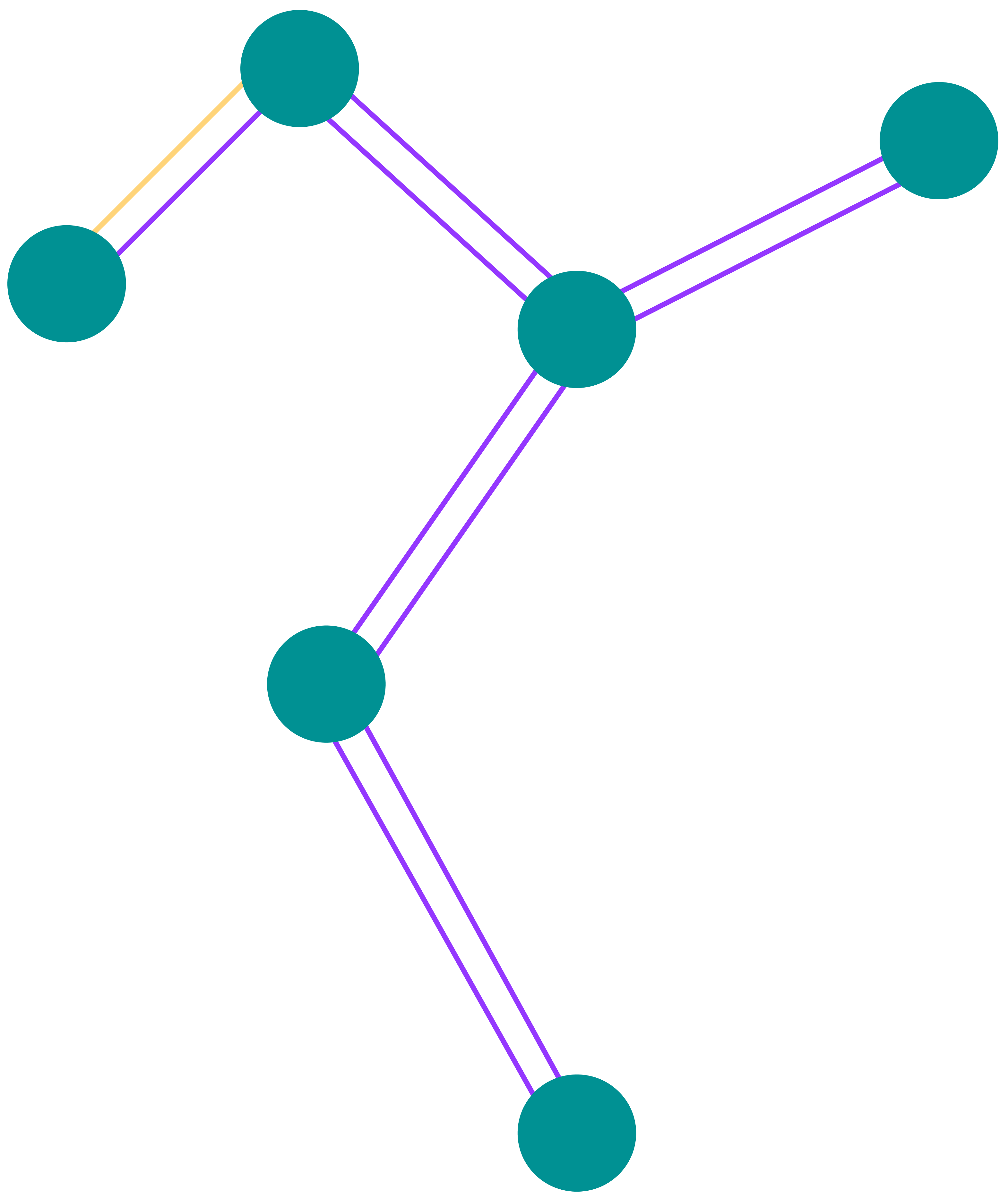


1. Find the MST T

2. Duplicate all edges of T

Metric TSP : 2-Approximation

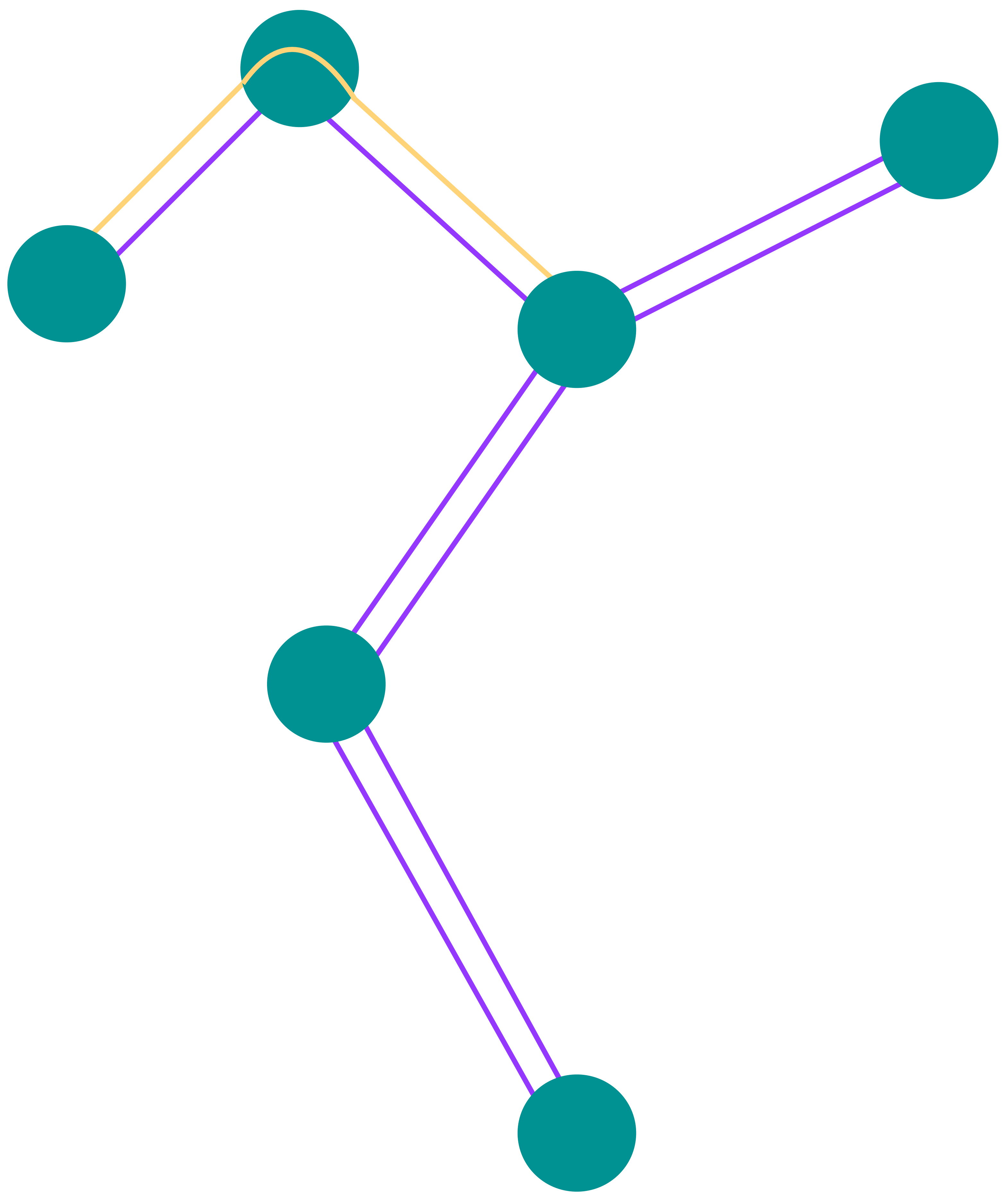
Algorithm



1. Find the MST T
2. Duplicate all edges of T
3. Find Eulerian Tour W

Metric TSP : 2-Approximation

Algorithm



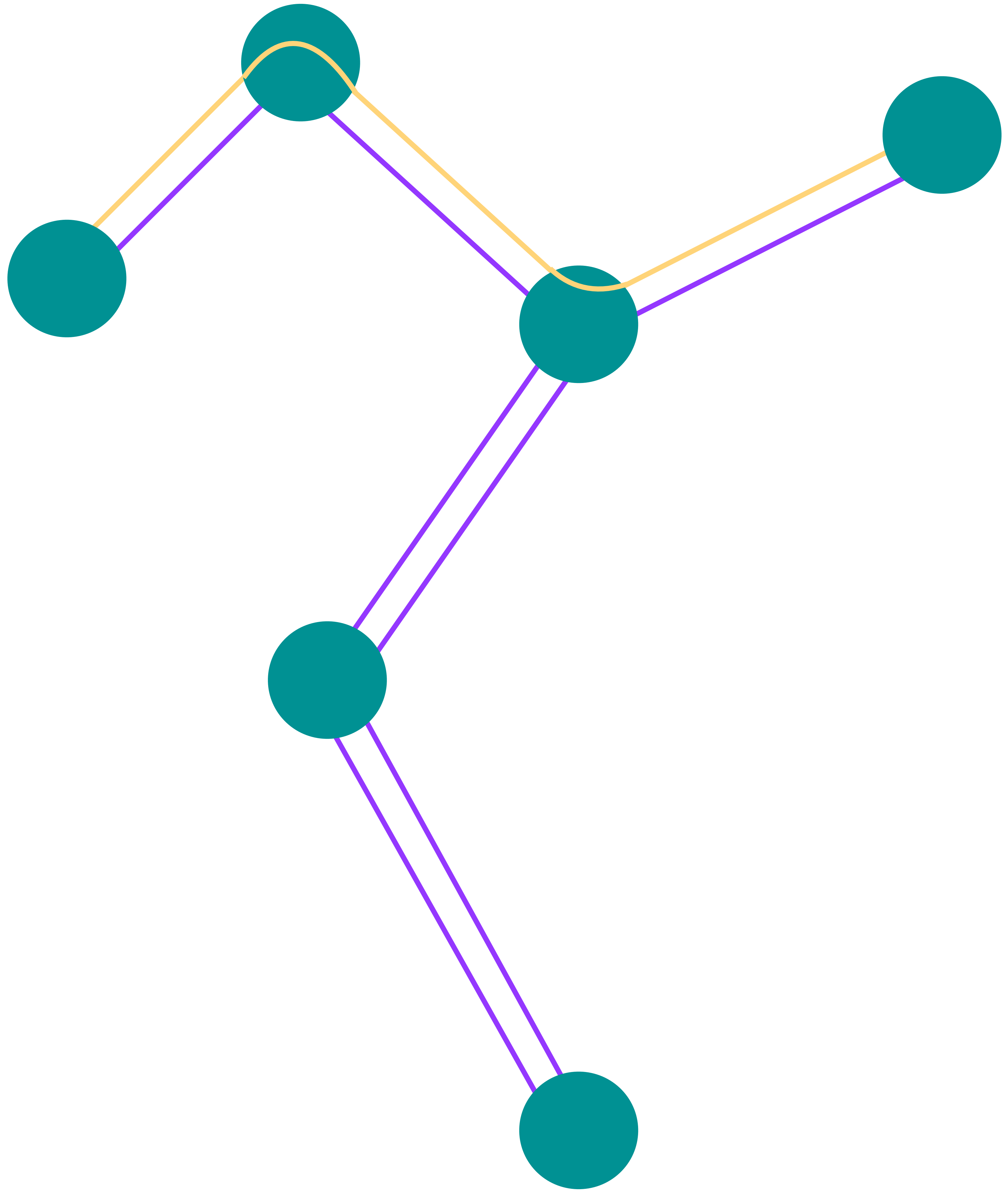
1. Find the MST T

2. Duplicate all edges of T

3. Find Eulerian Tour W

Metric TSP : 2-Approximation

Algorithm



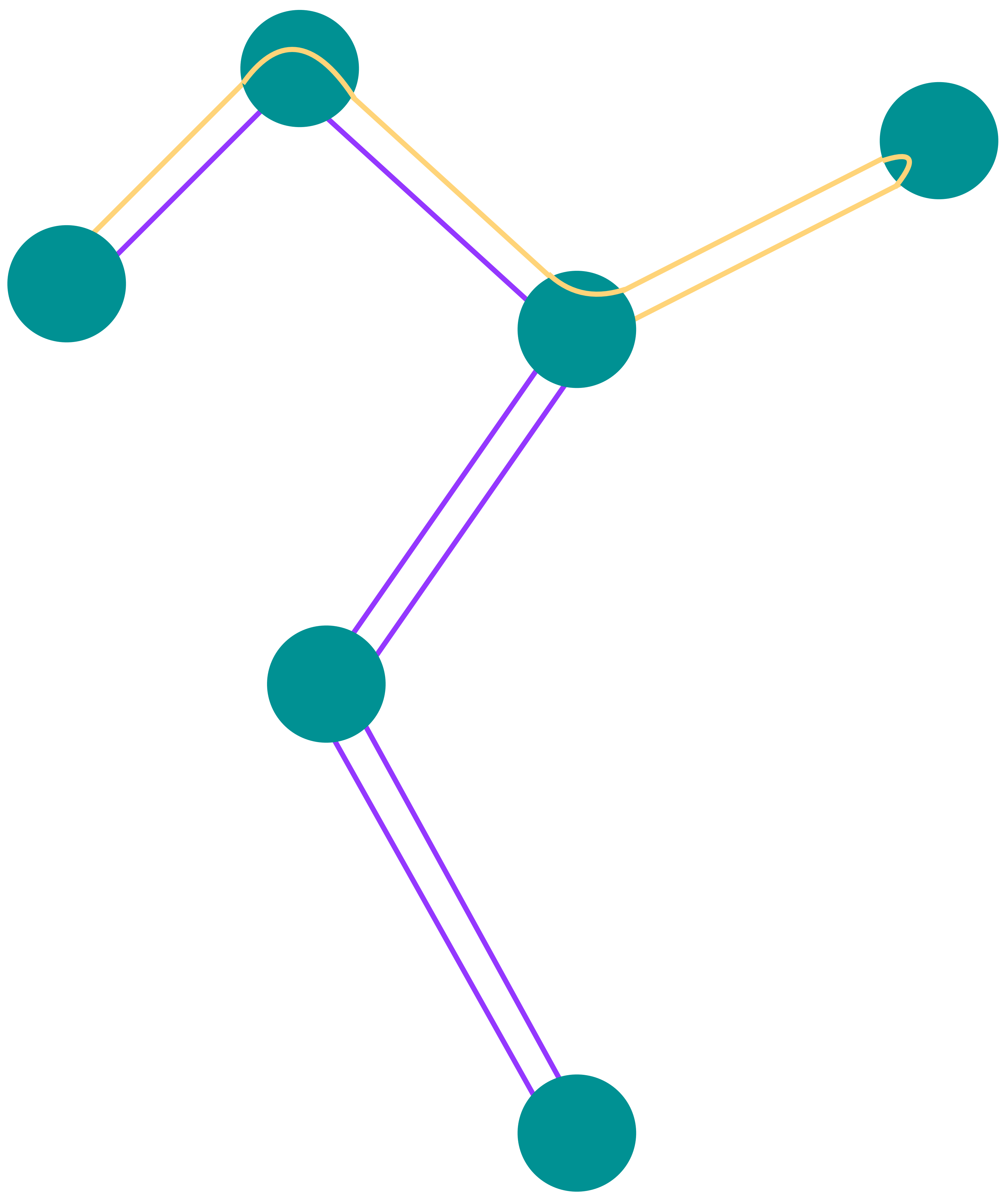
1. Find the MST T

2. Duplicate all edges of T

3. Find Eulerian Tour W

Metric TSP : 2-Approximation

Algorithm



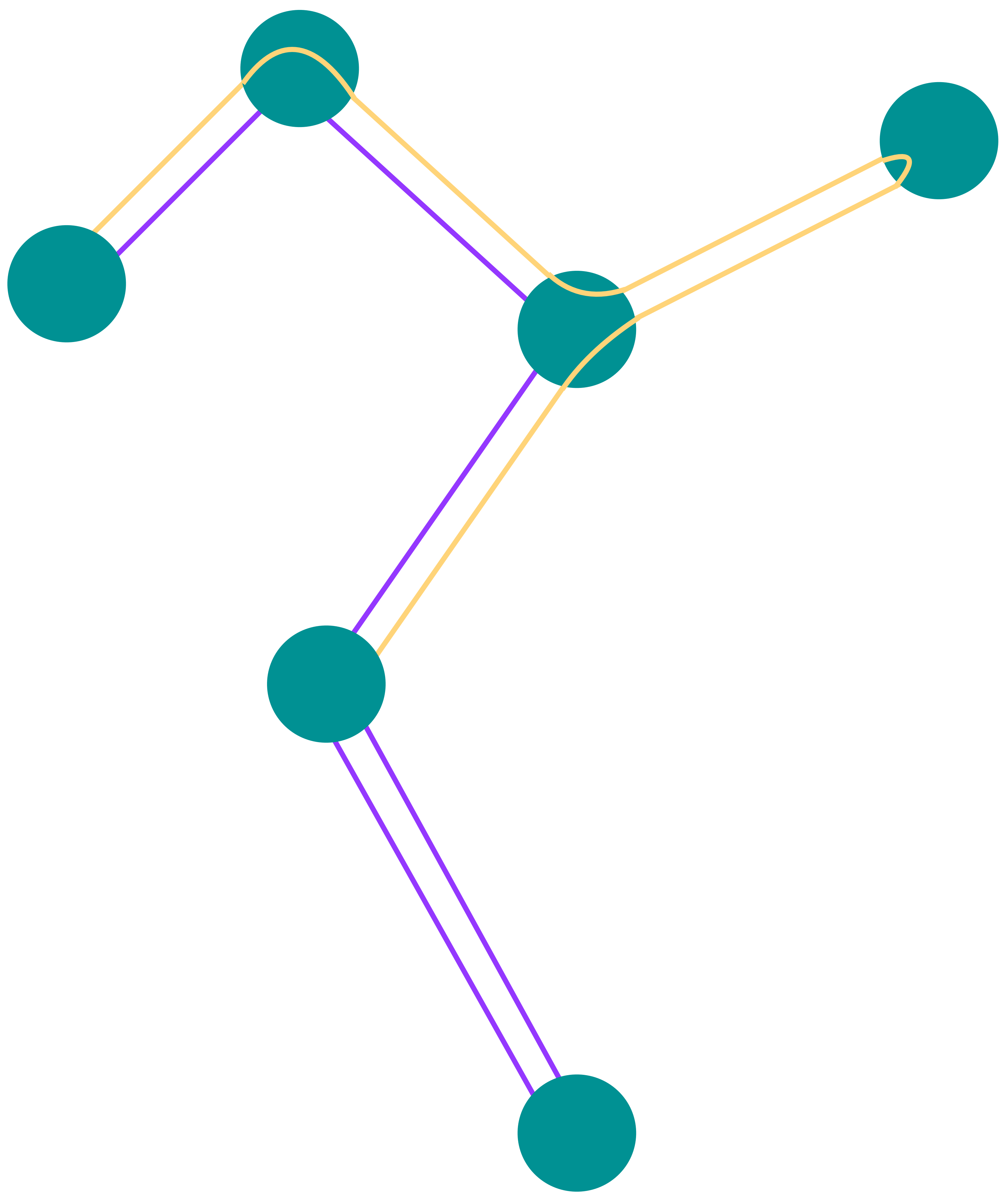
1. Find the MST T

2. Duplicate all edges of T

3. Find Eulerian Tour W

Metric TSP : 2-Approximation

Algorithm



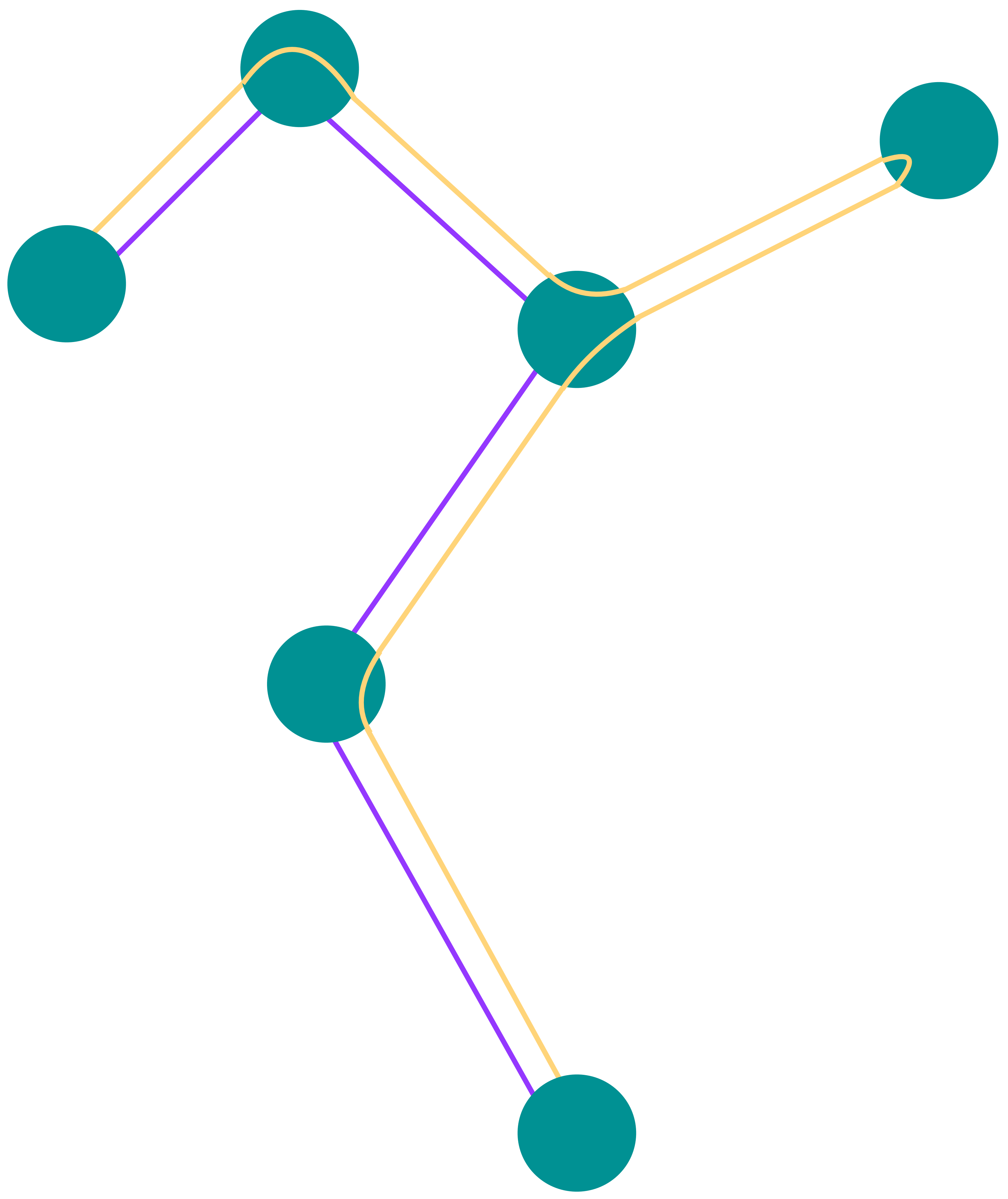
1. Find the MST T

2. Duplicate all edges of T

3. Find Eulerian Tour W

Metric TSP : 2-Approximation

Algorithm



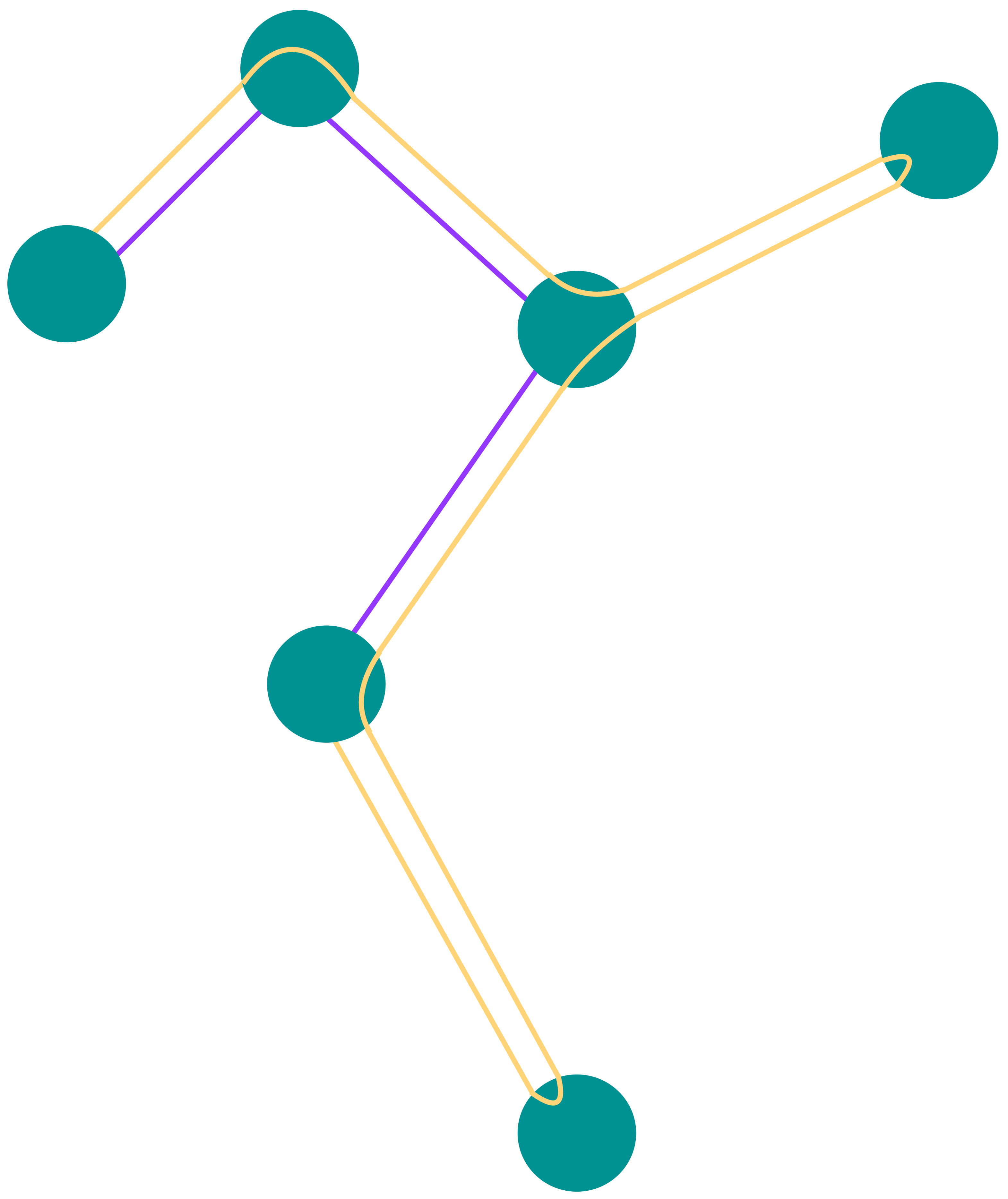
1. Find the MST T

2. Duplicate all edges of T

3. Find Eulerian Tour W

Metric TSP : 2-Approximation

Algorithm



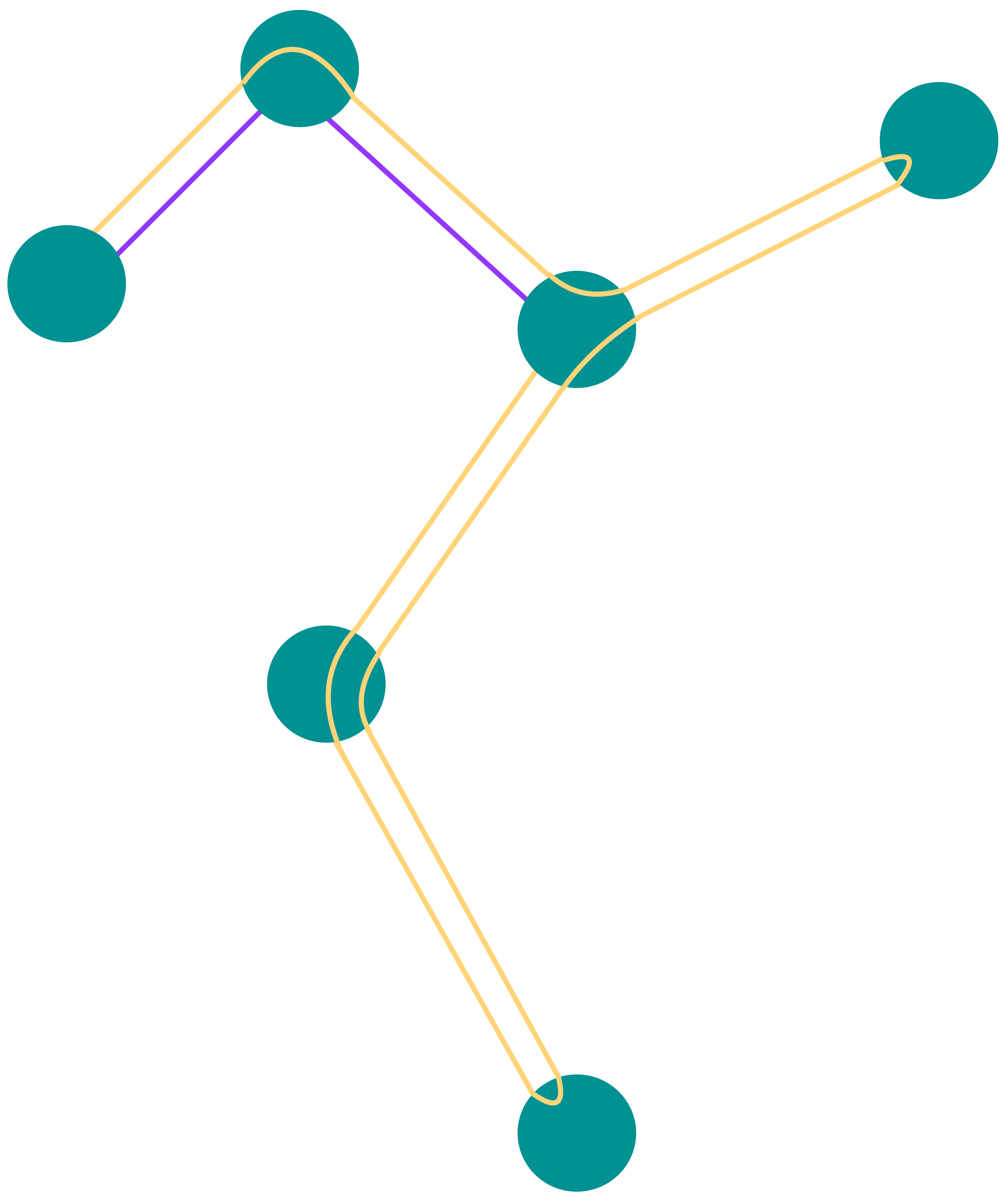
1. Find the MST T

2. Duplicate all edges of T

3. Find Eulerian Tour W

Metric TSP : 2-Approximation

Algorithm



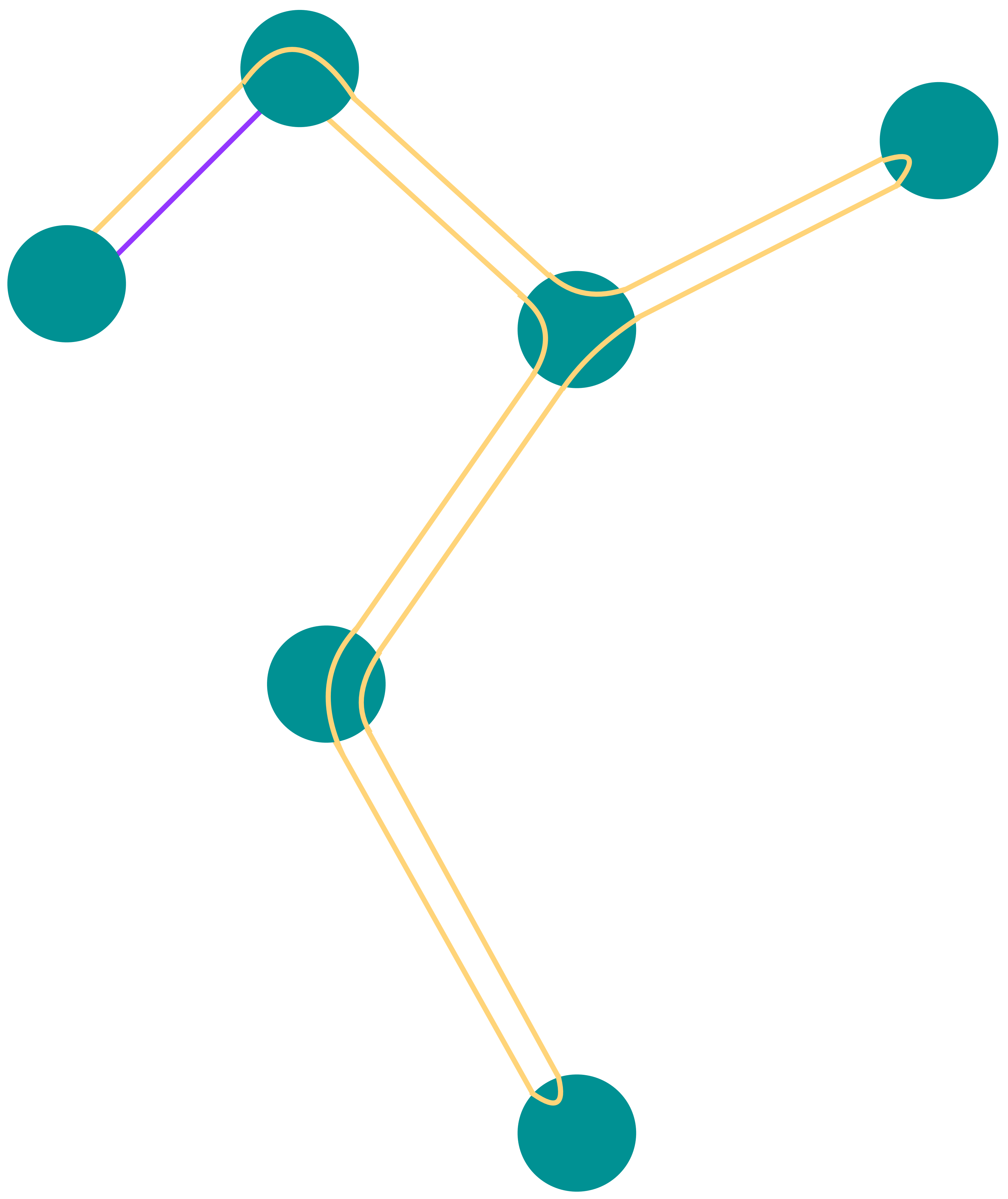
1. Find the MST T

2. Duplicate all edges of T

3. Find Eulerian Tour W

Metric TSP : 2-Approximation

Algorithm



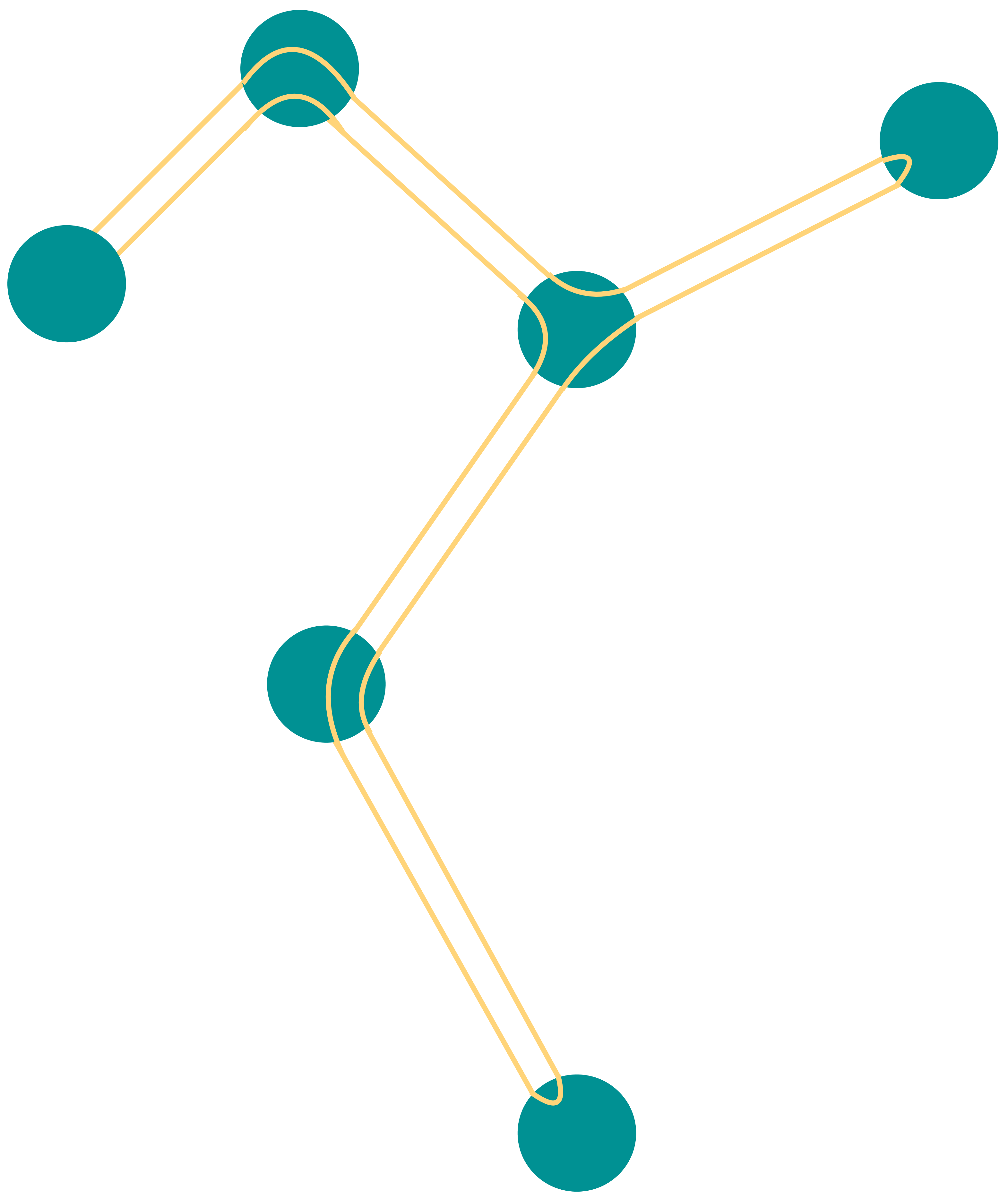
1. Find the MST T

2. Duplicate all edges of T

3. Find Eulerian Tour W

Metric TSP : 2-Approximation

Algorithm



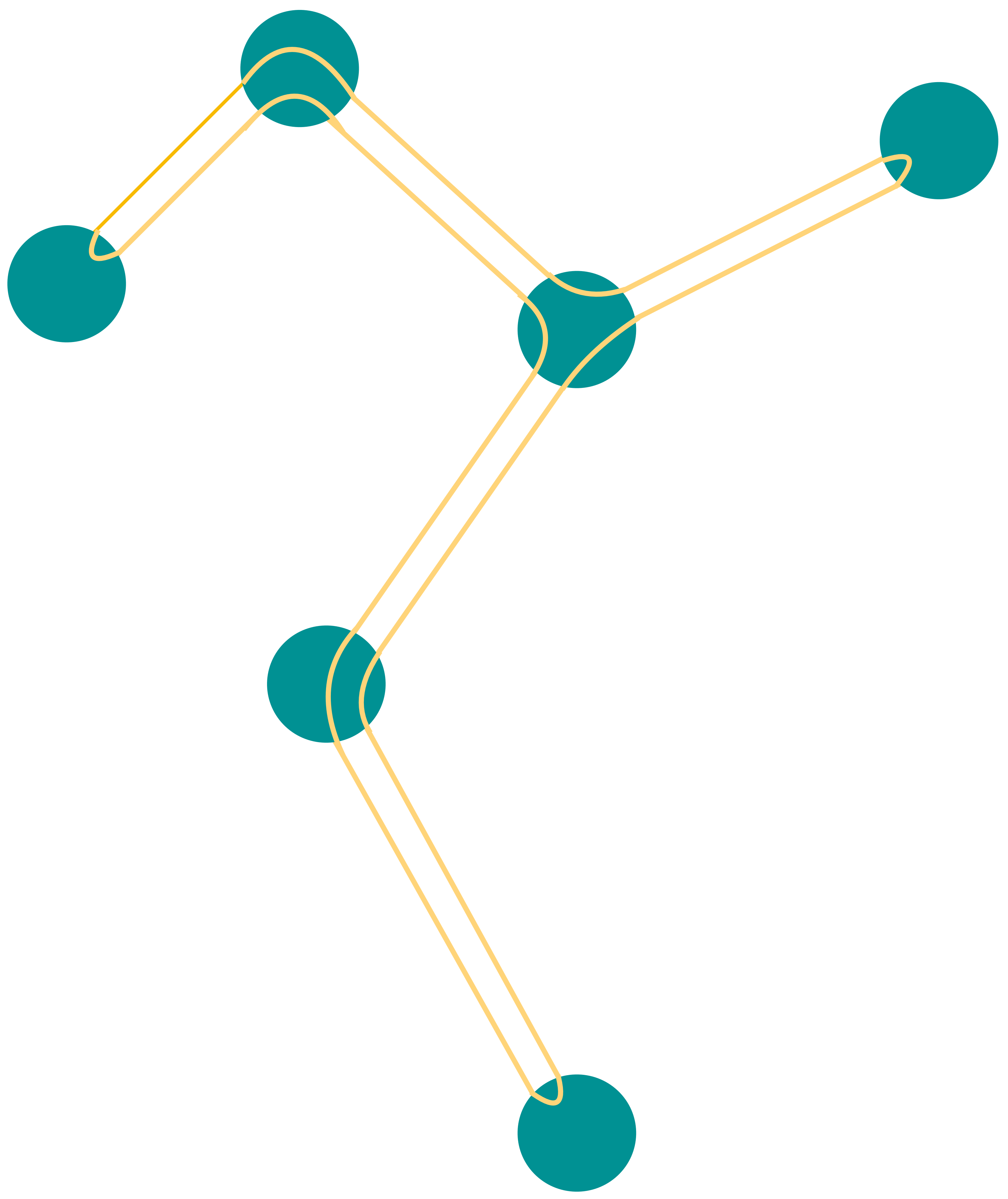
1. Find the MST T

2. Duplicate all edges of T

3. Find Eulerian Tour W

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

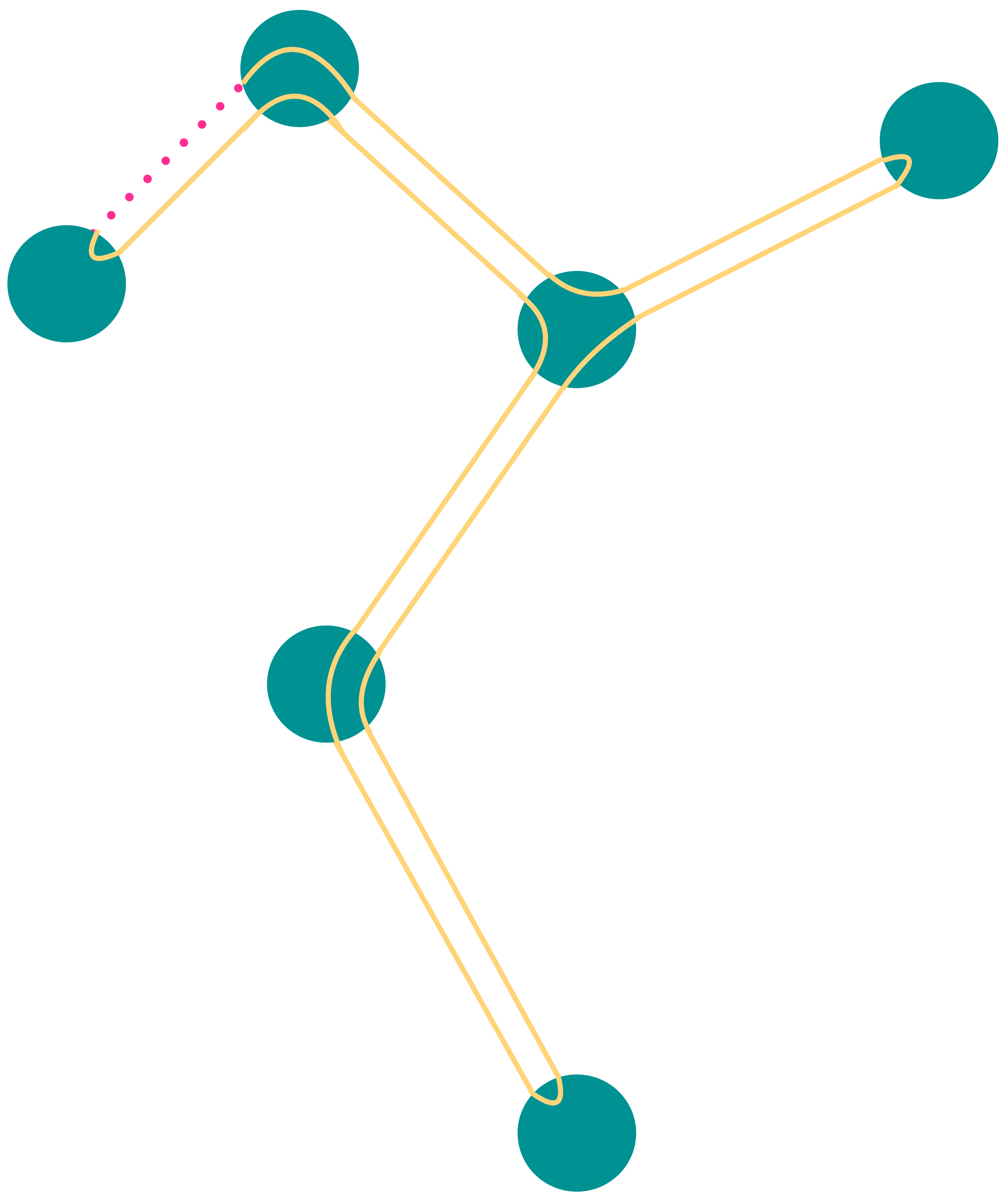
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

⇒ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

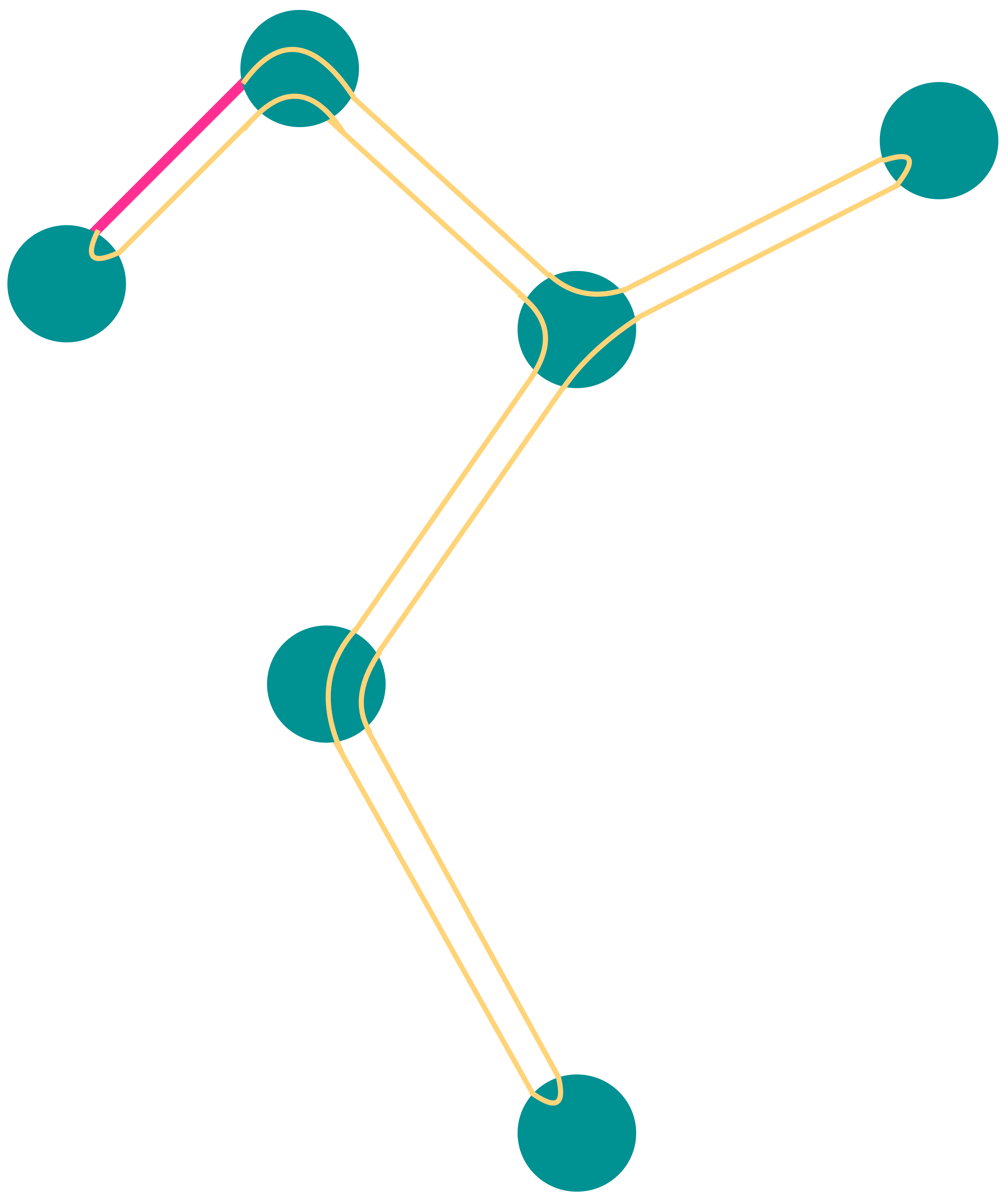
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

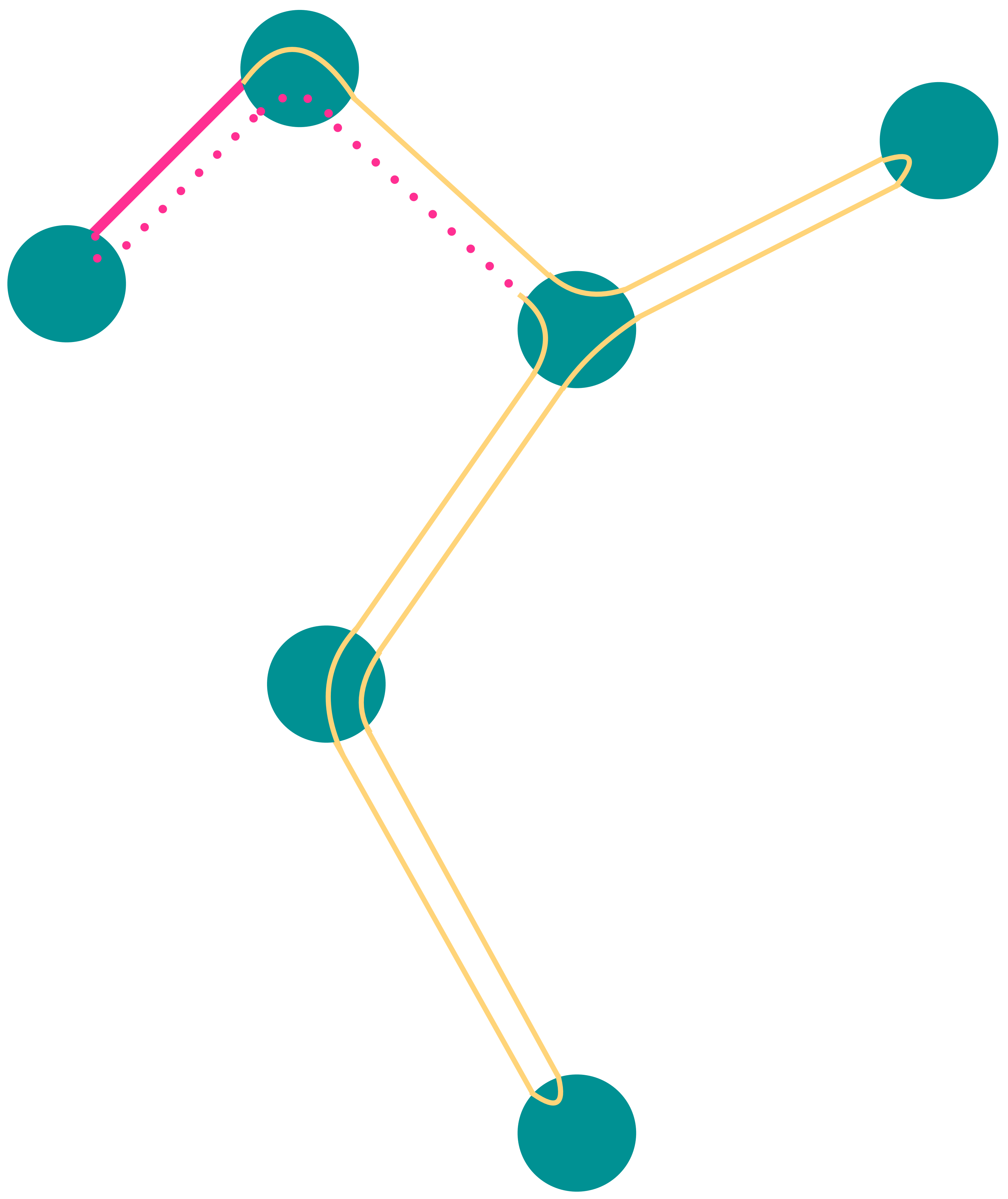
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

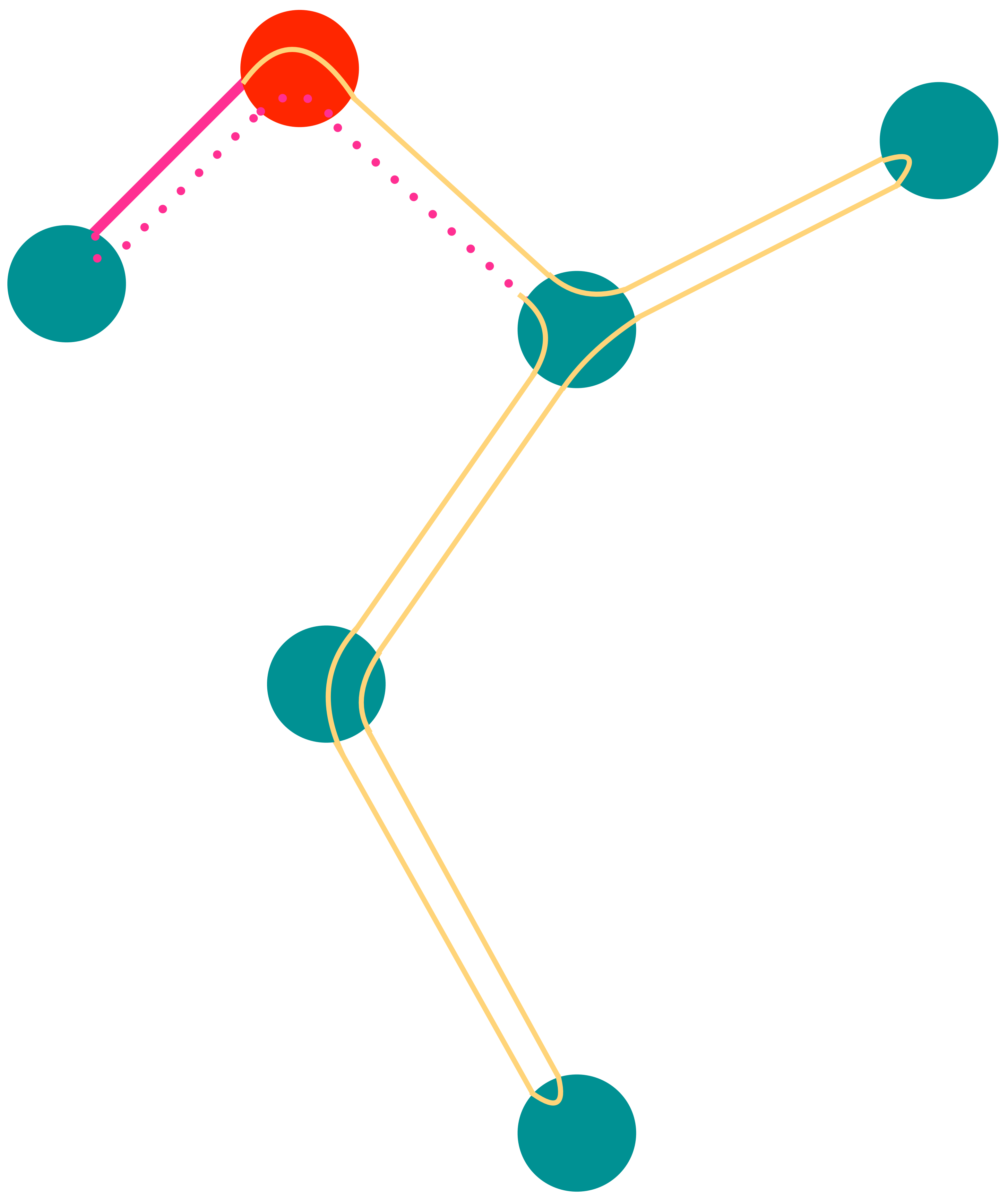
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

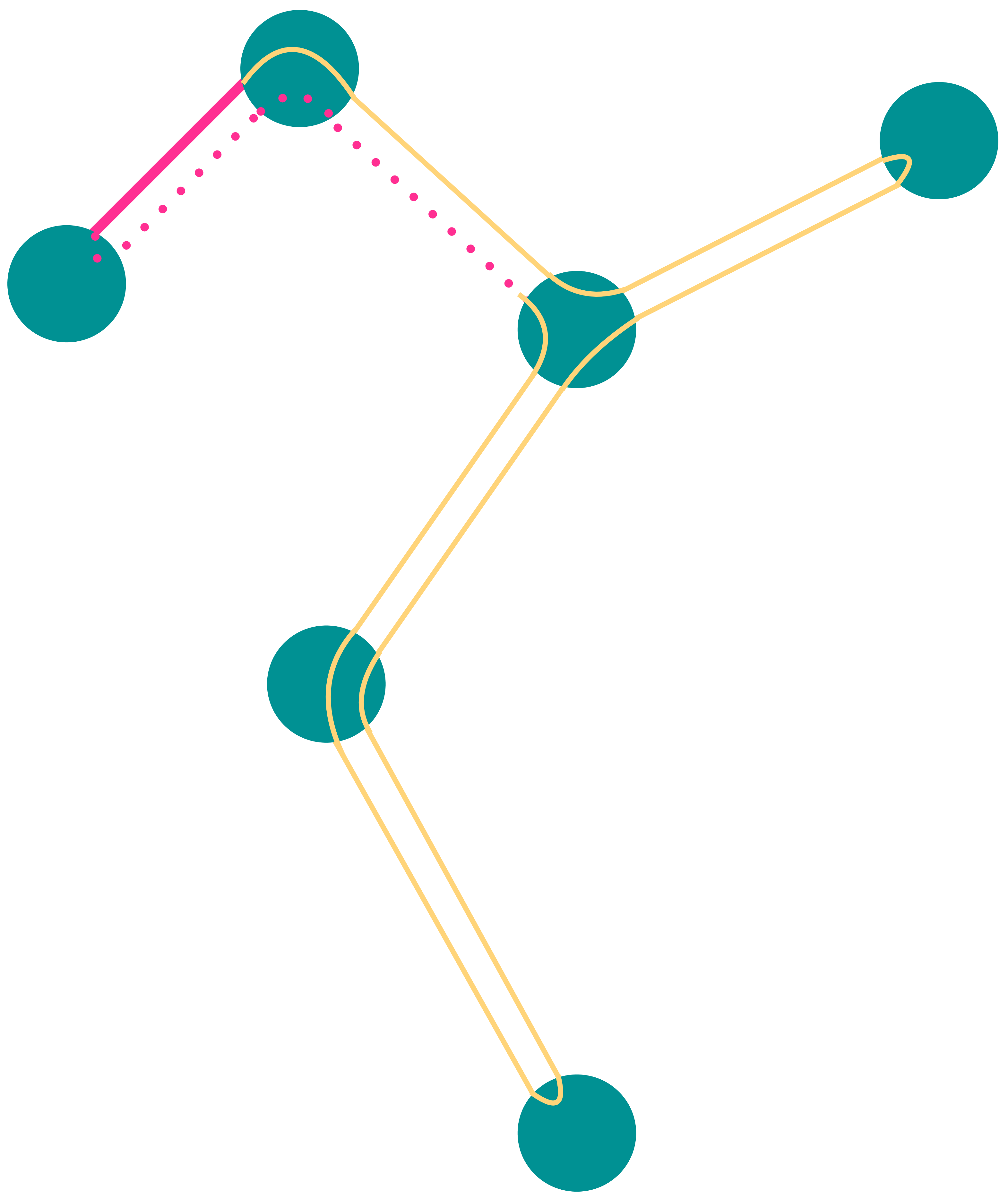
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

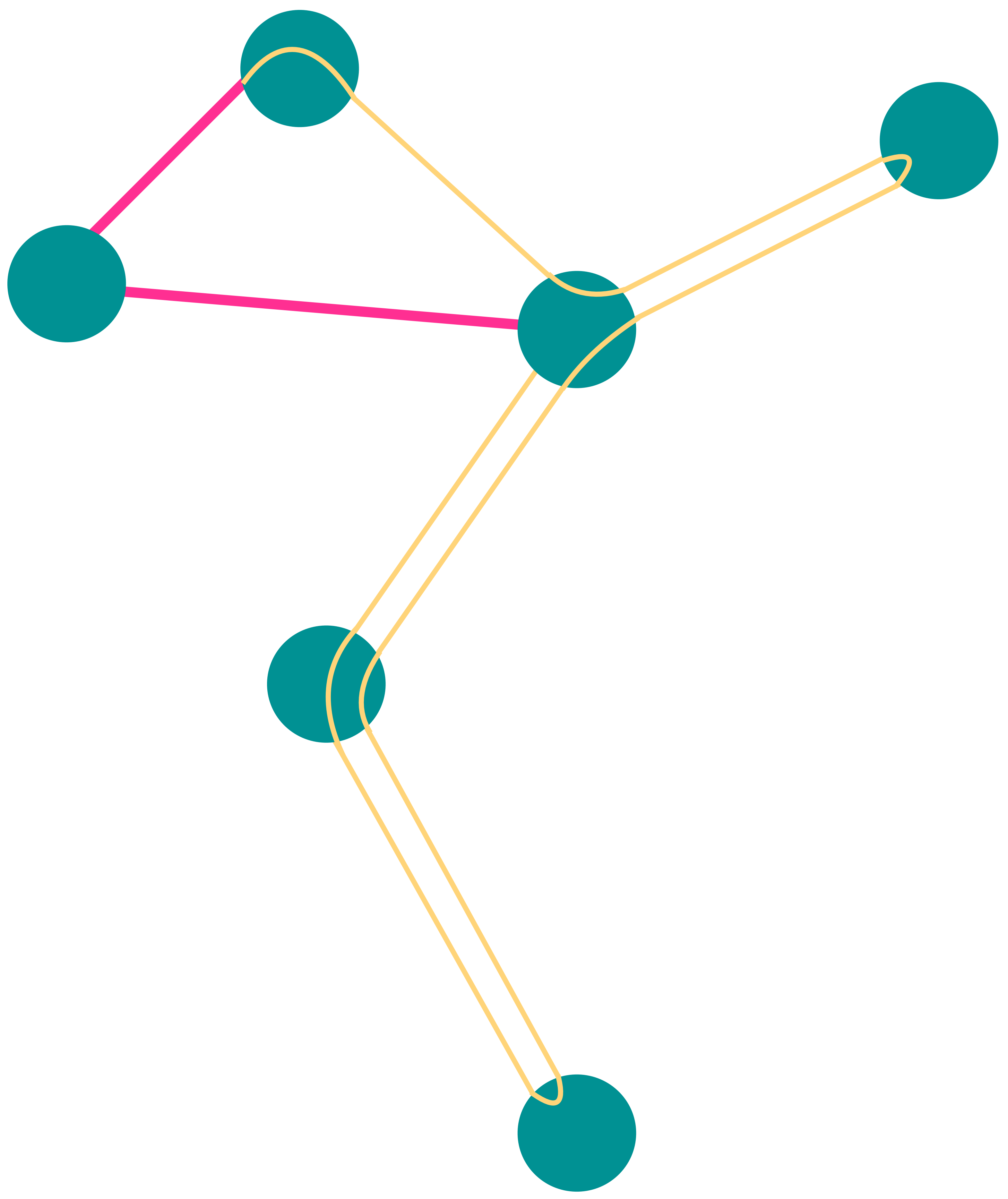
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

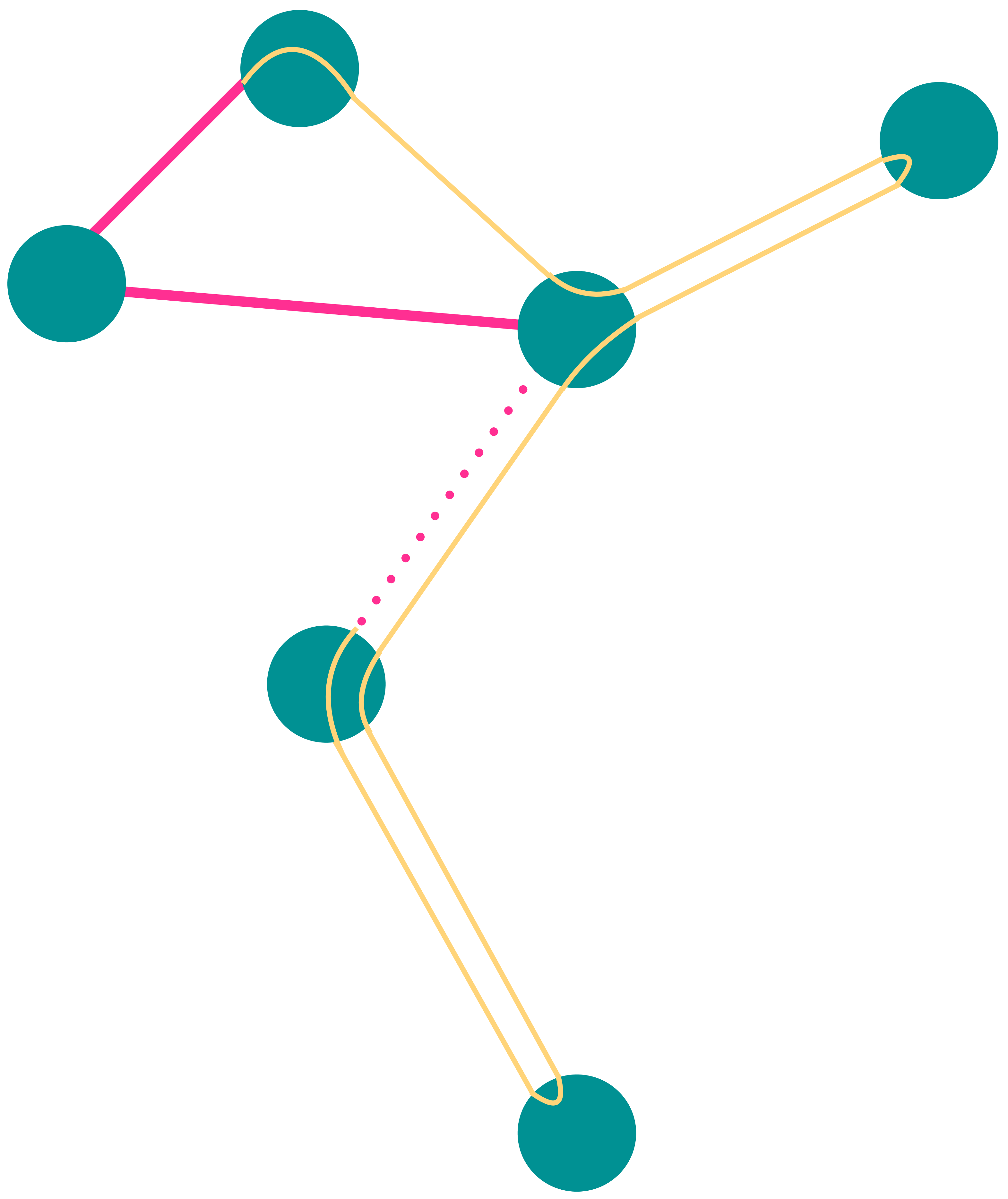
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

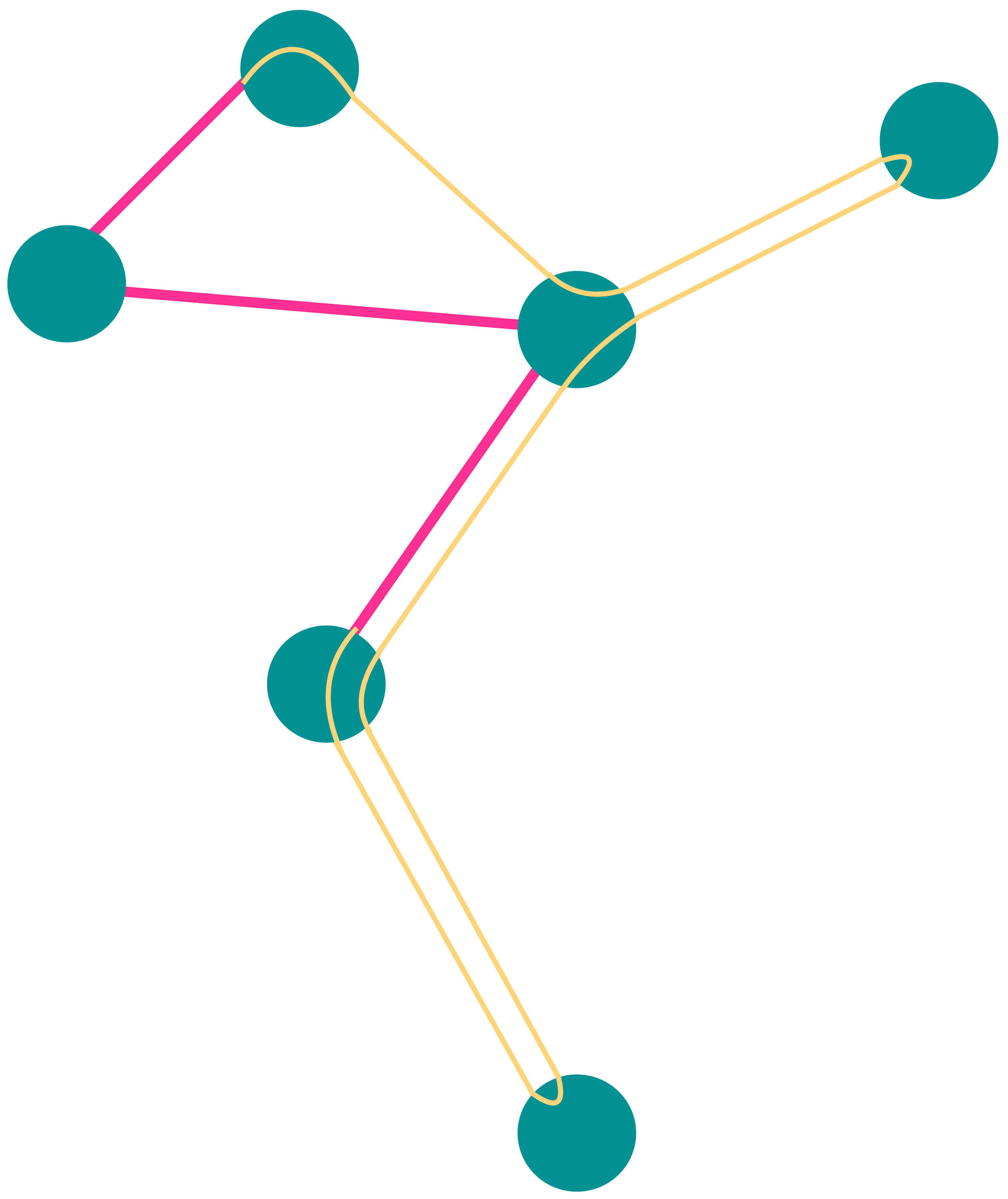
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

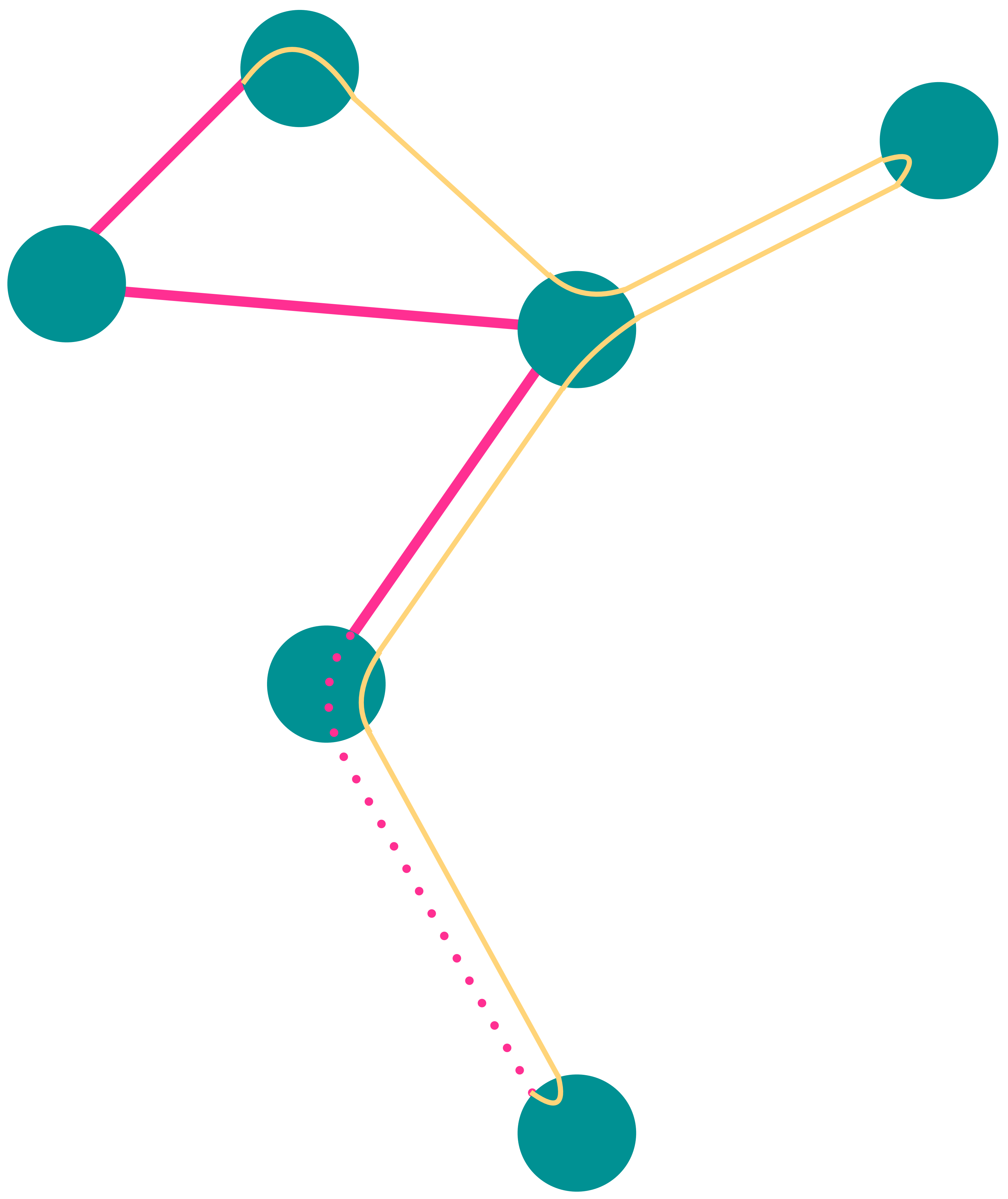
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

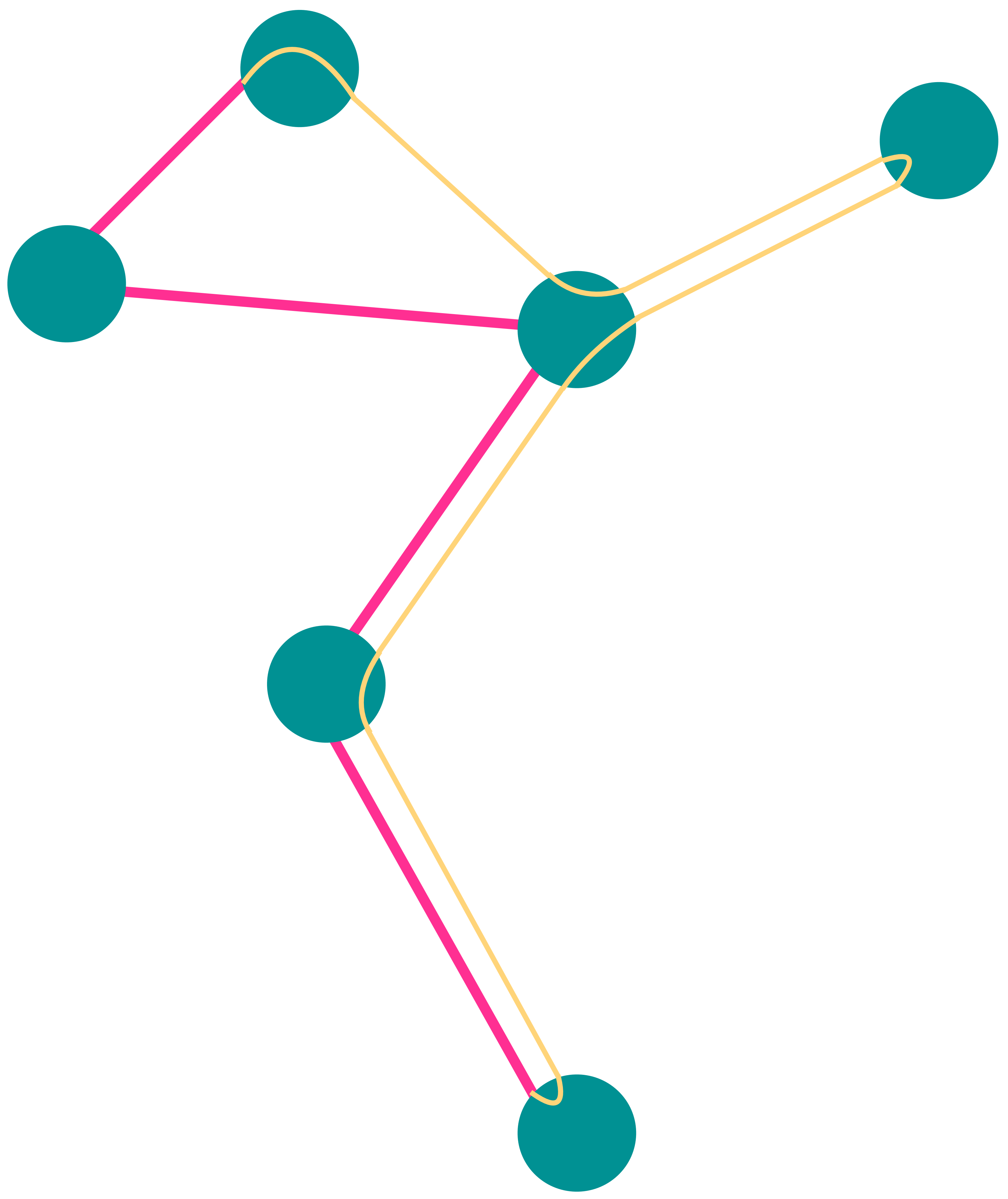
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

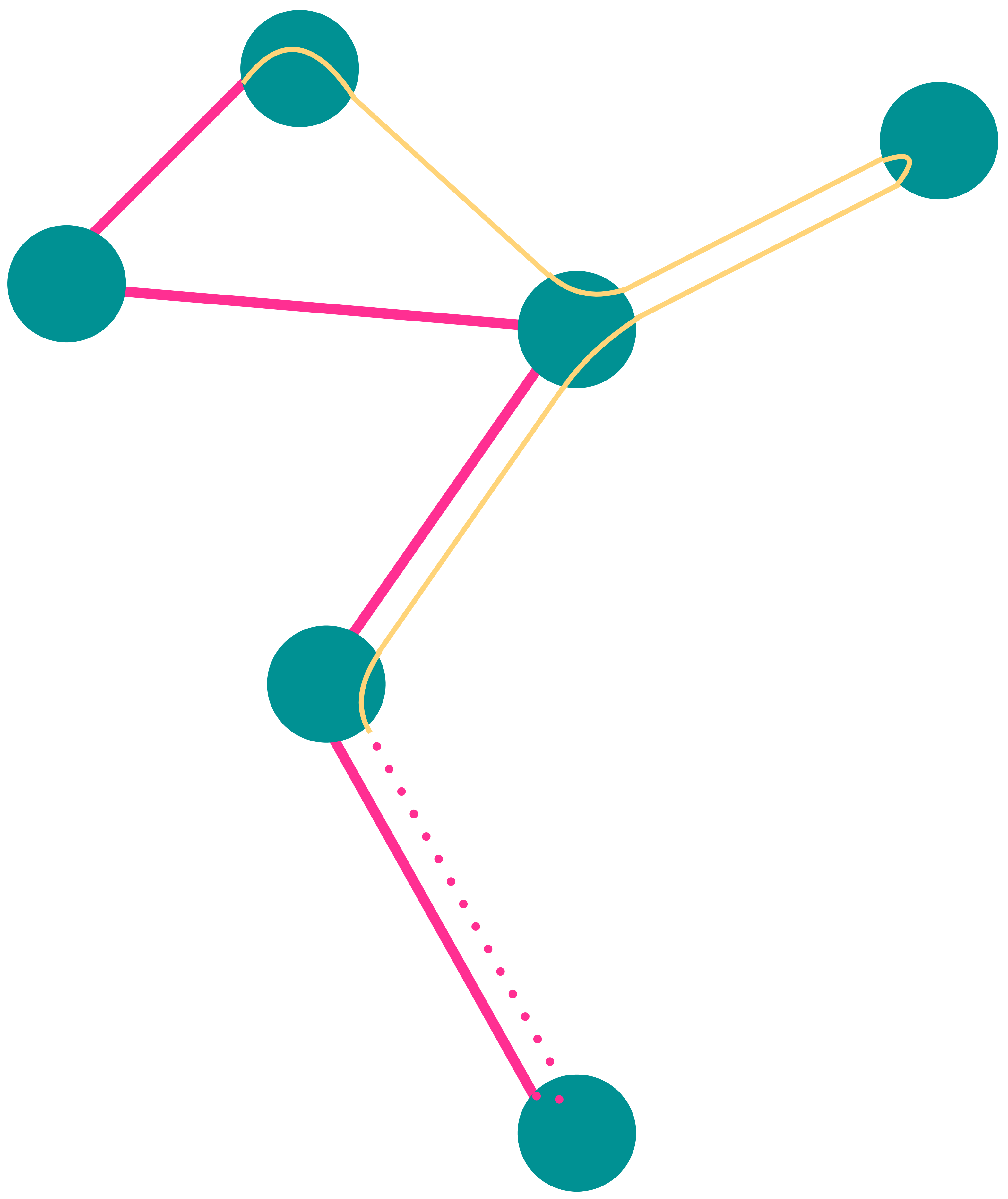
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

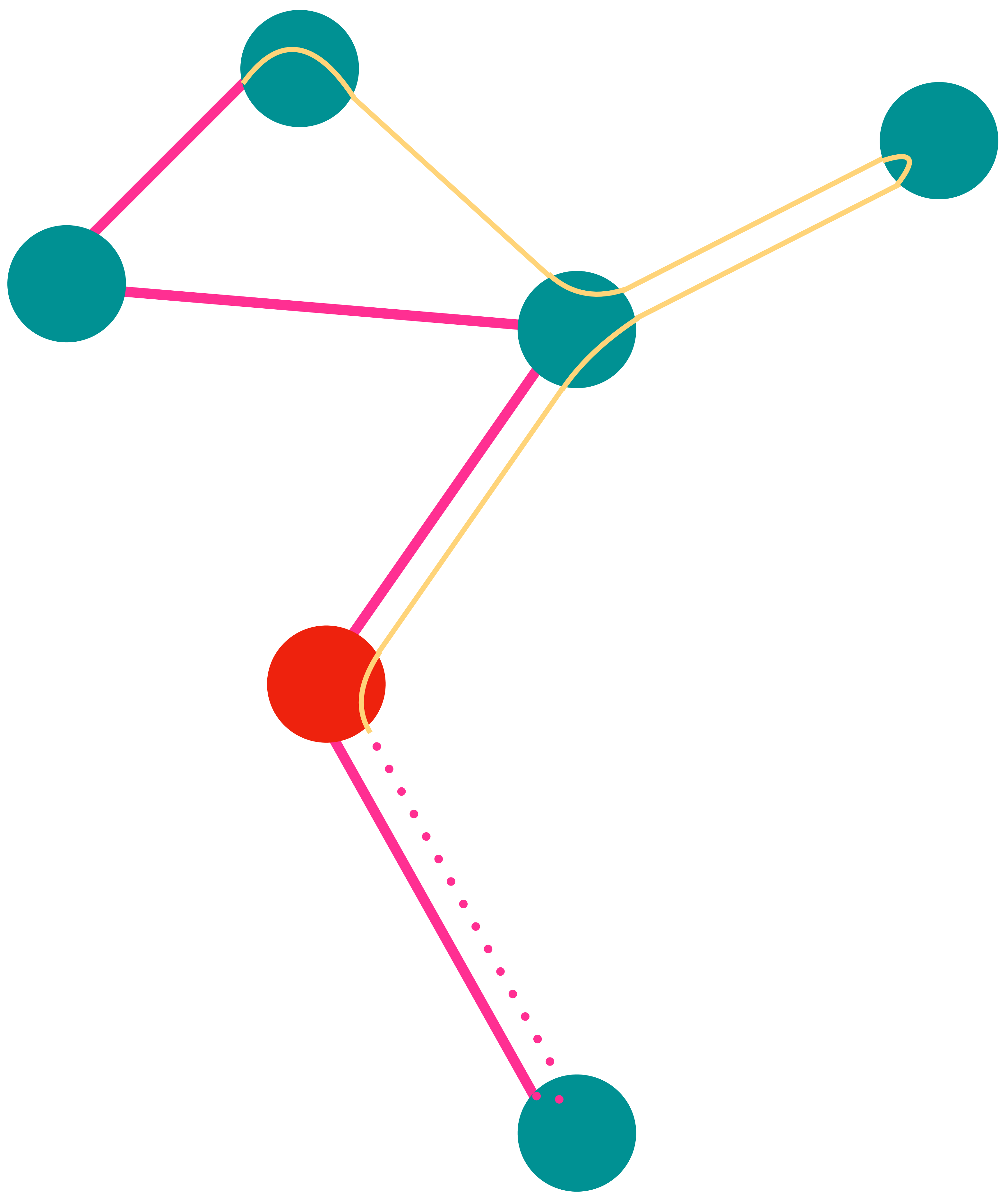
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

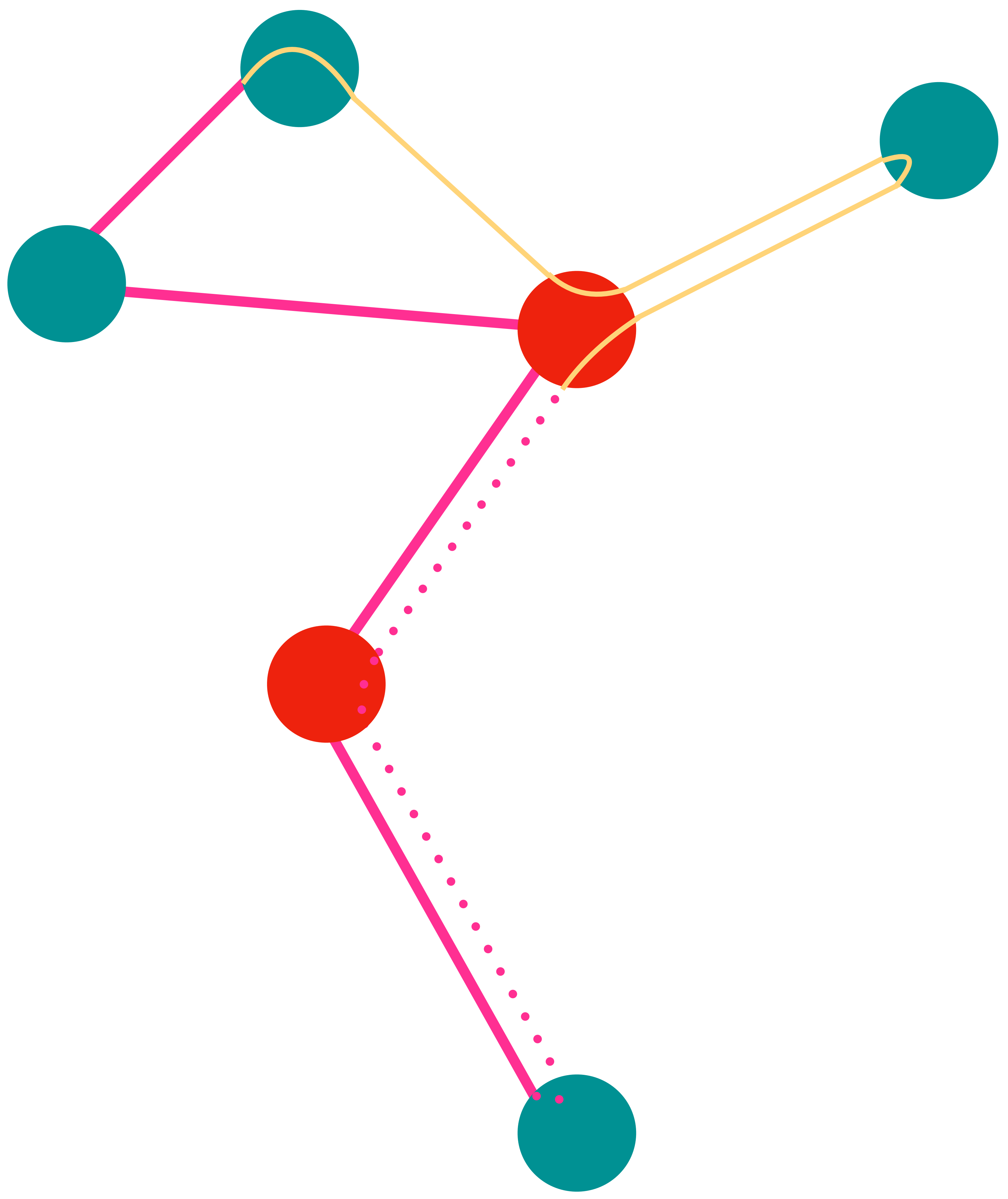
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

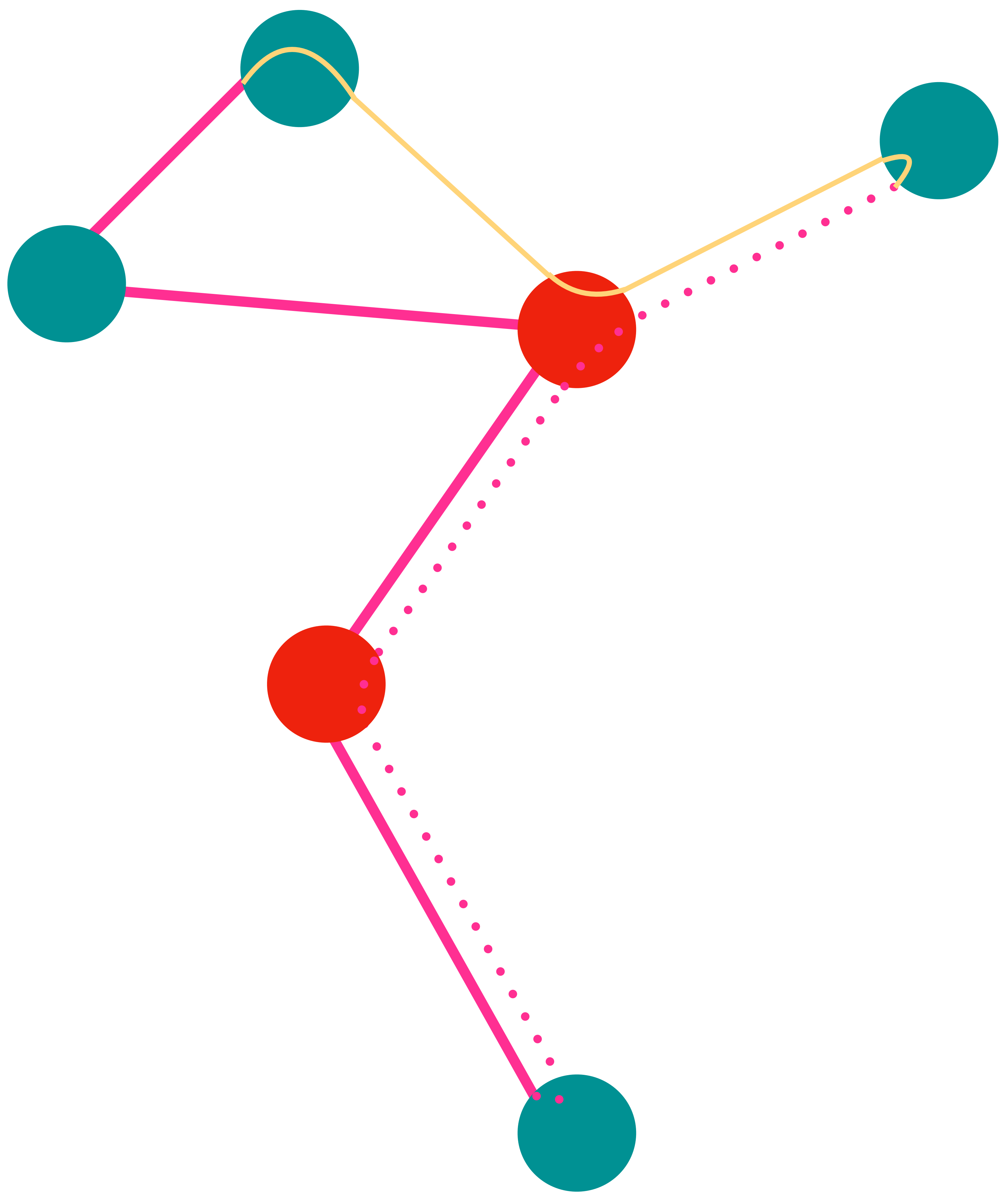
2. Duplicate all edges of T

3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation
Algorithm



1. Find the MST T

2. Duplicate all edges of T

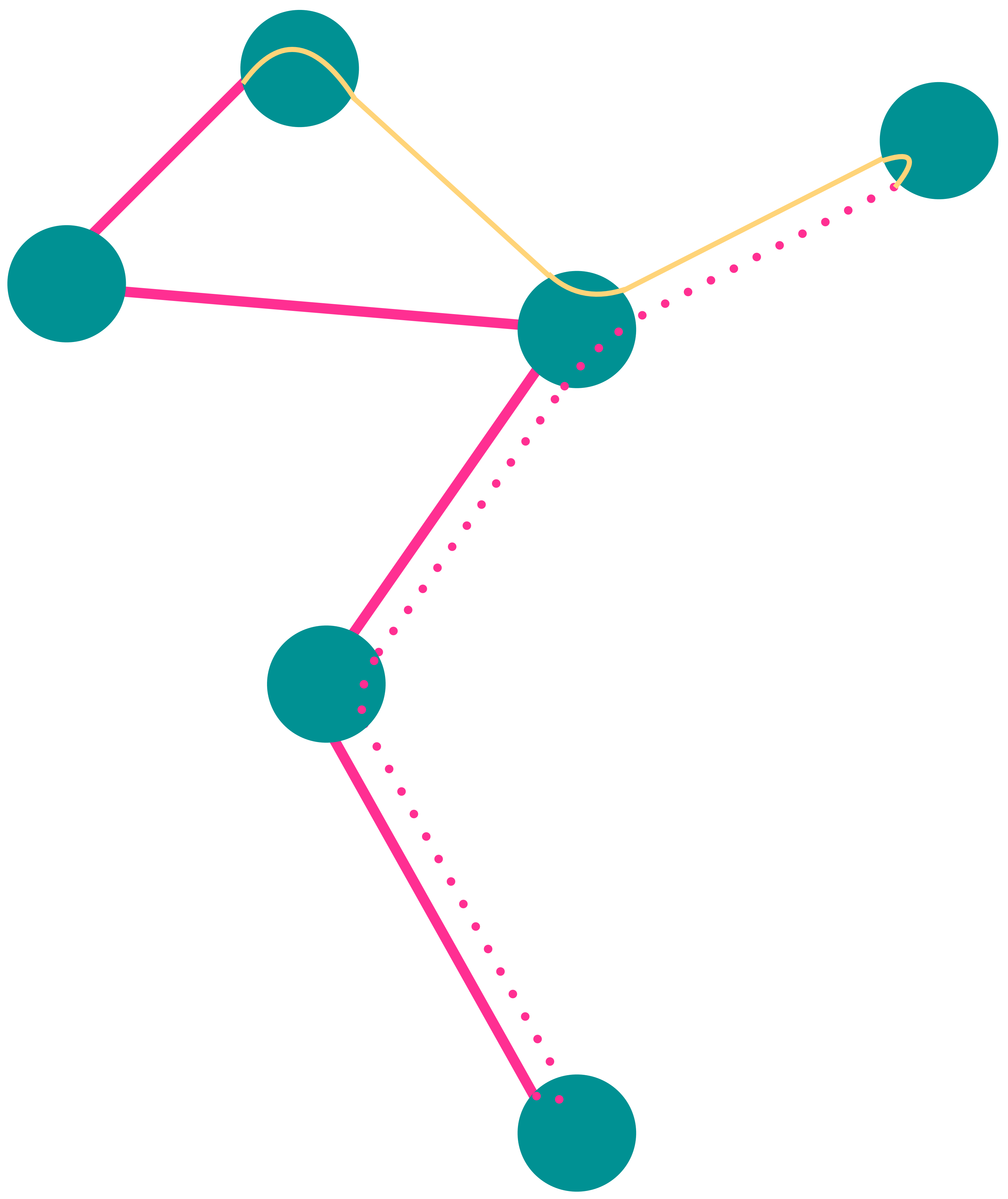
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

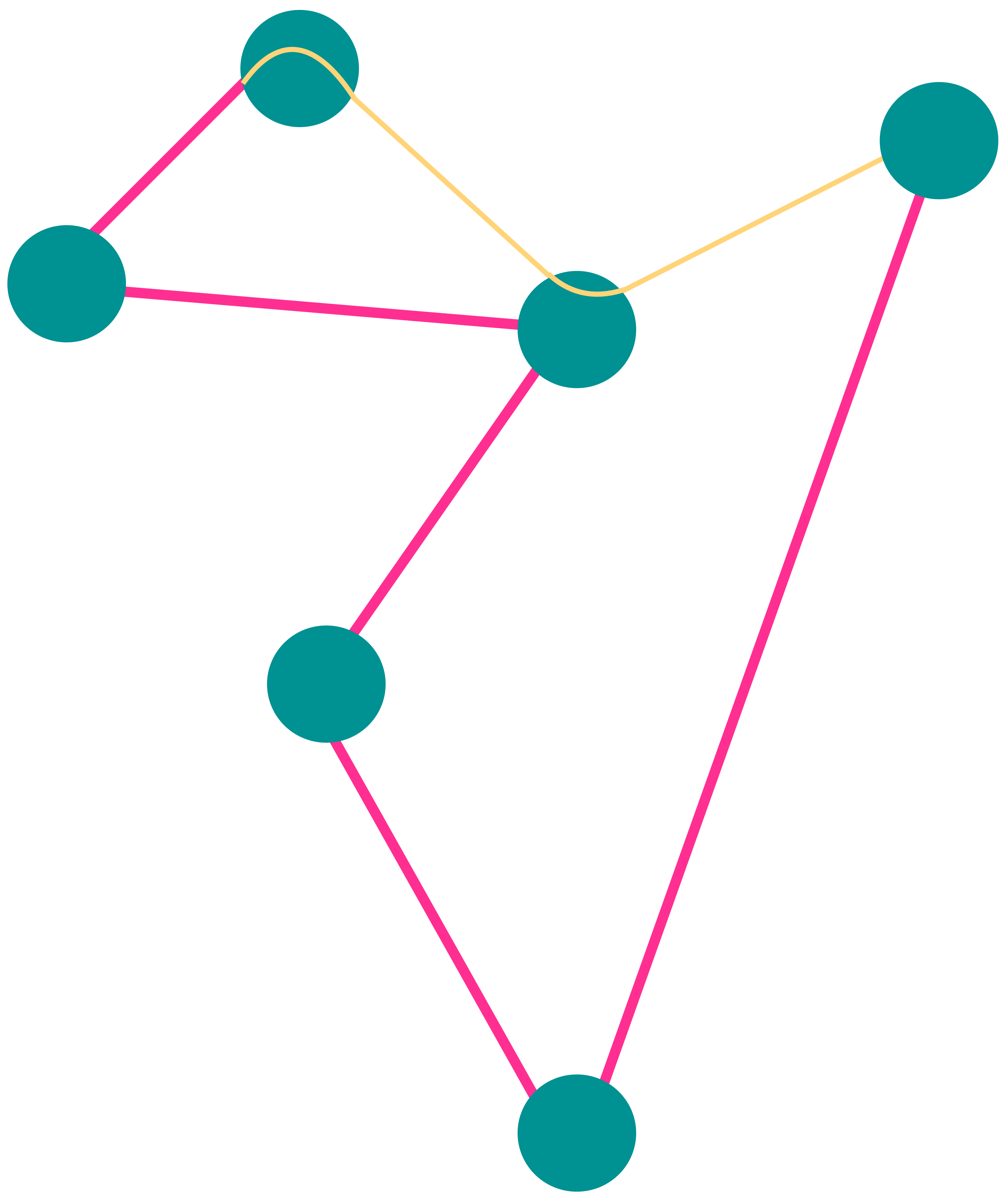
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

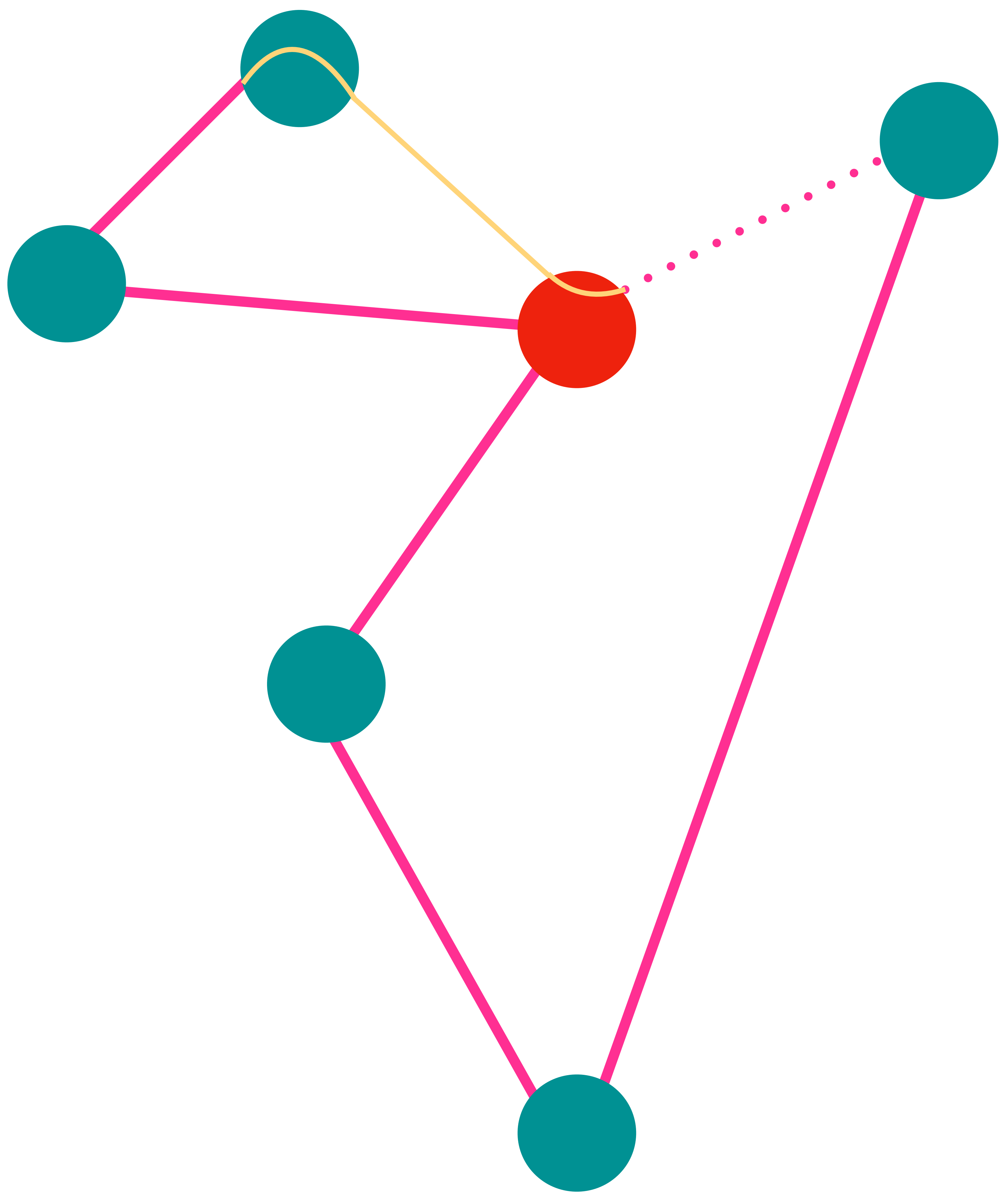
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

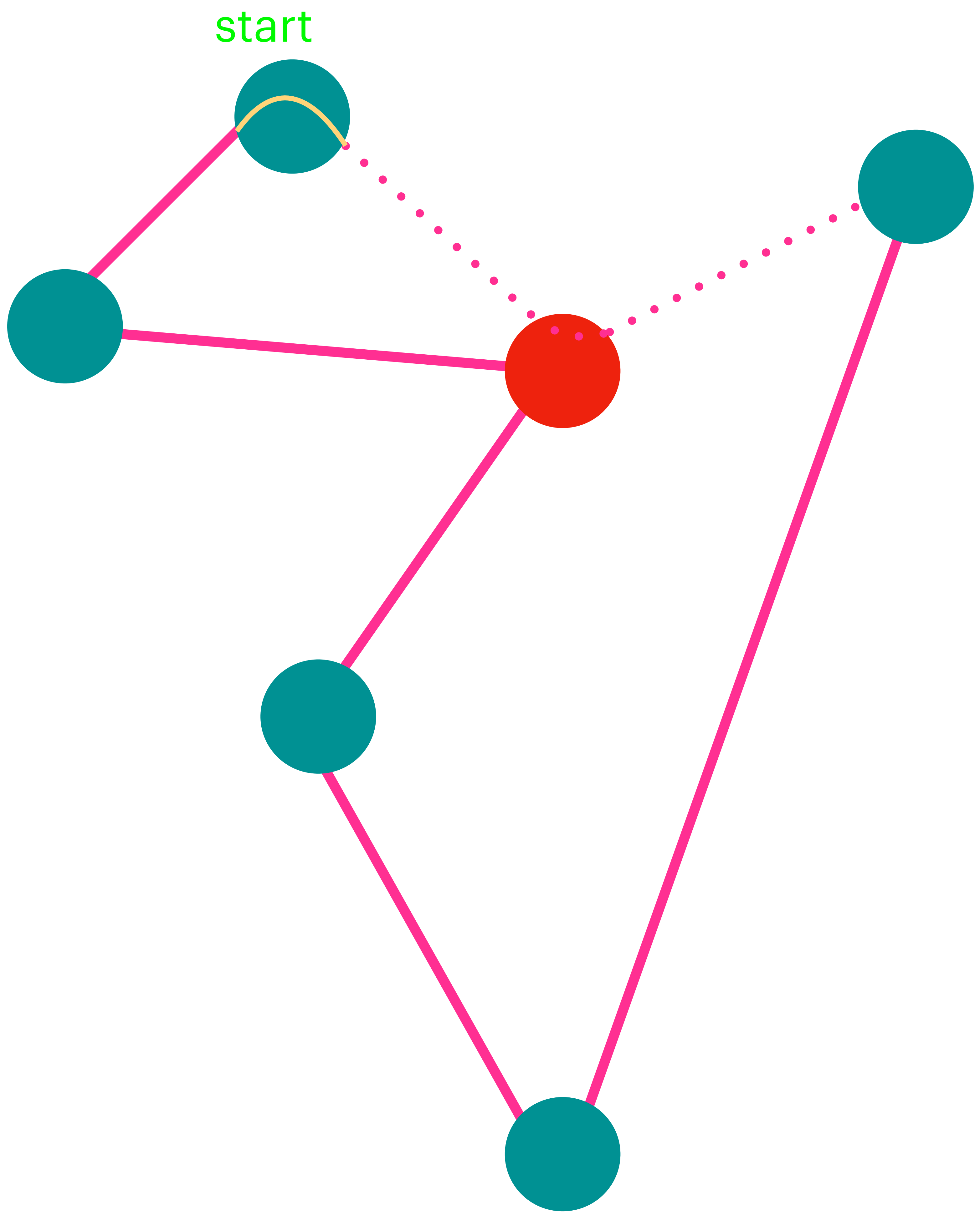
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

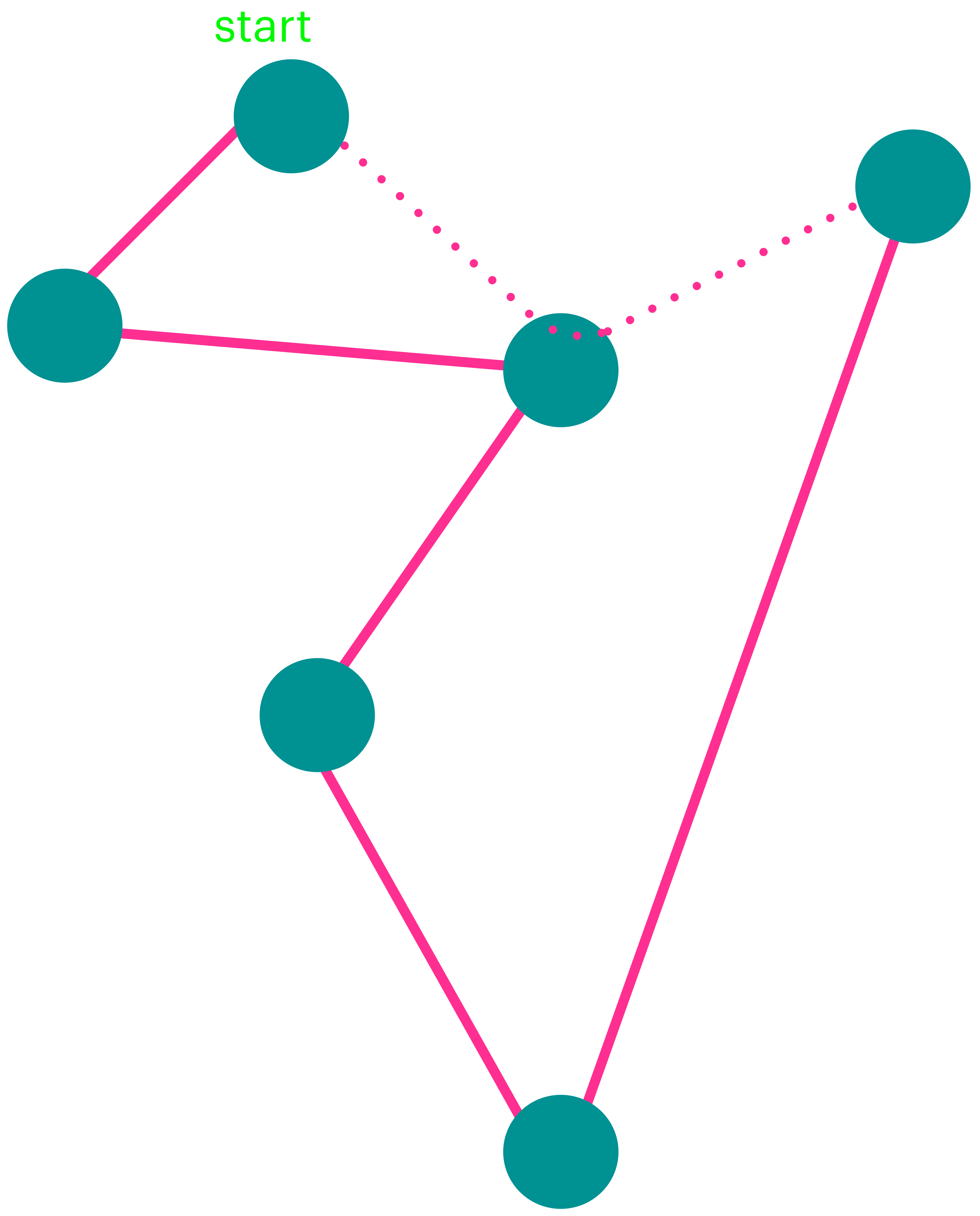
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

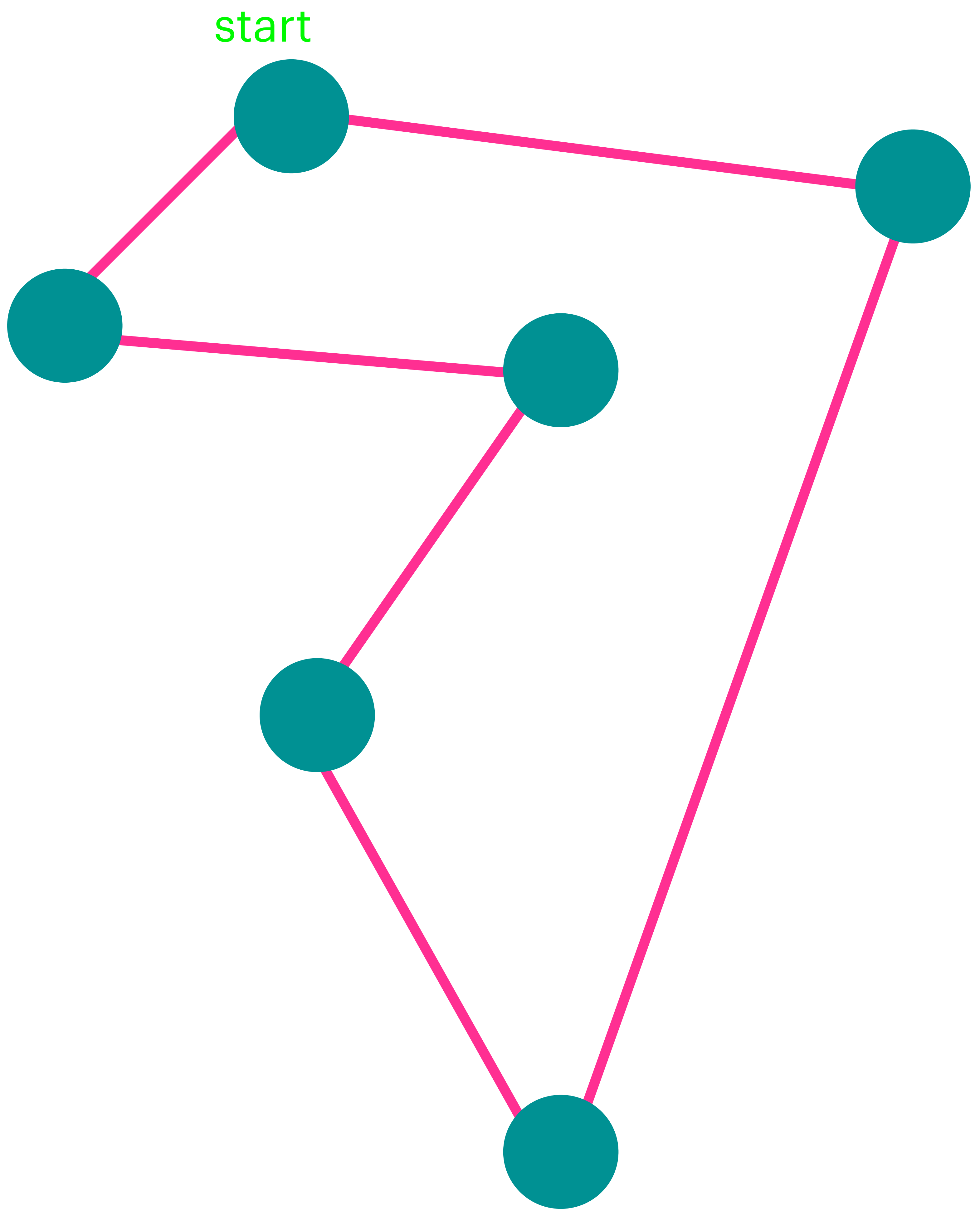
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

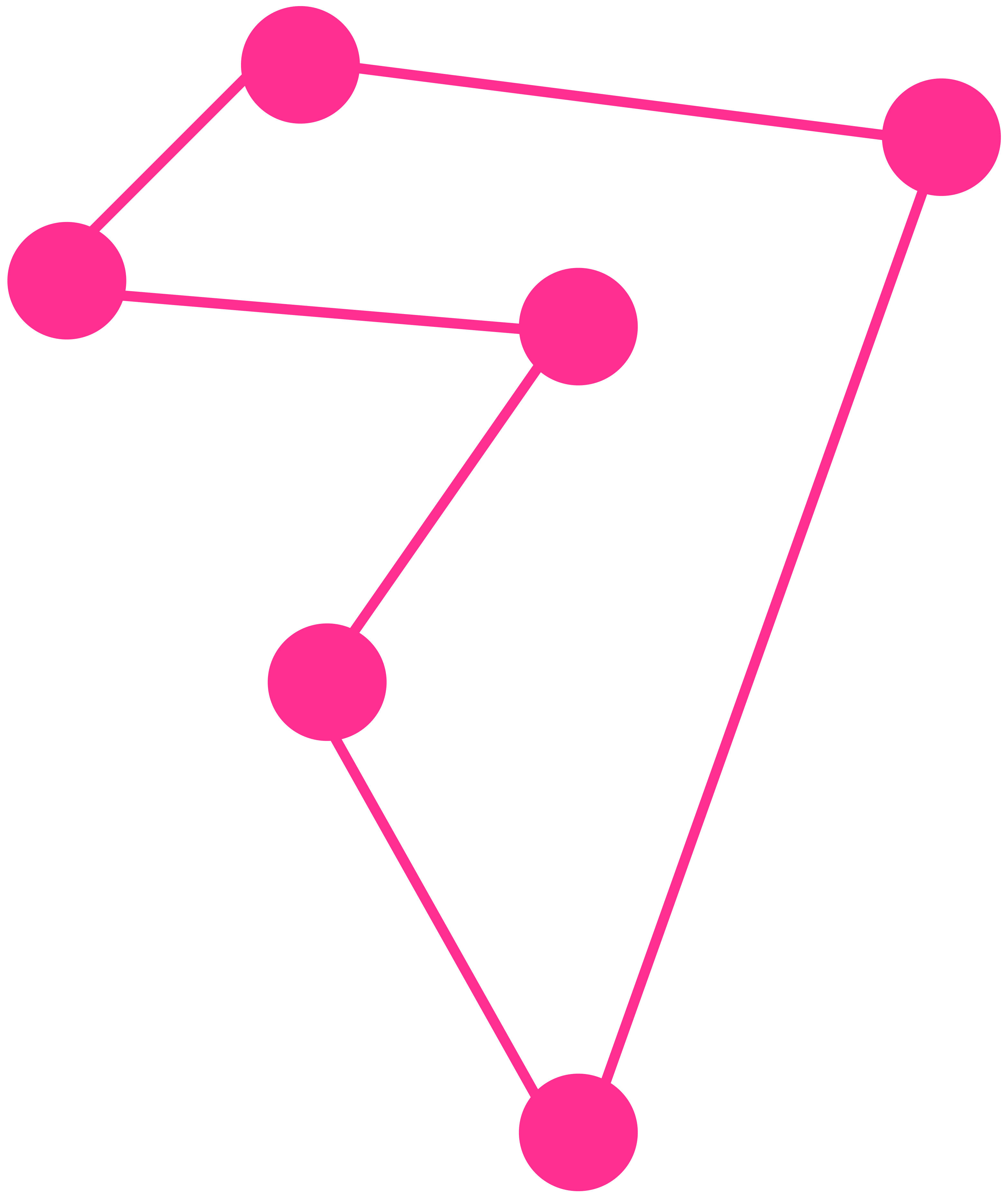
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

Metric TSP : 2-Approximation

Algorithm



1. Find the MST T

2. Duplicate all edges of T

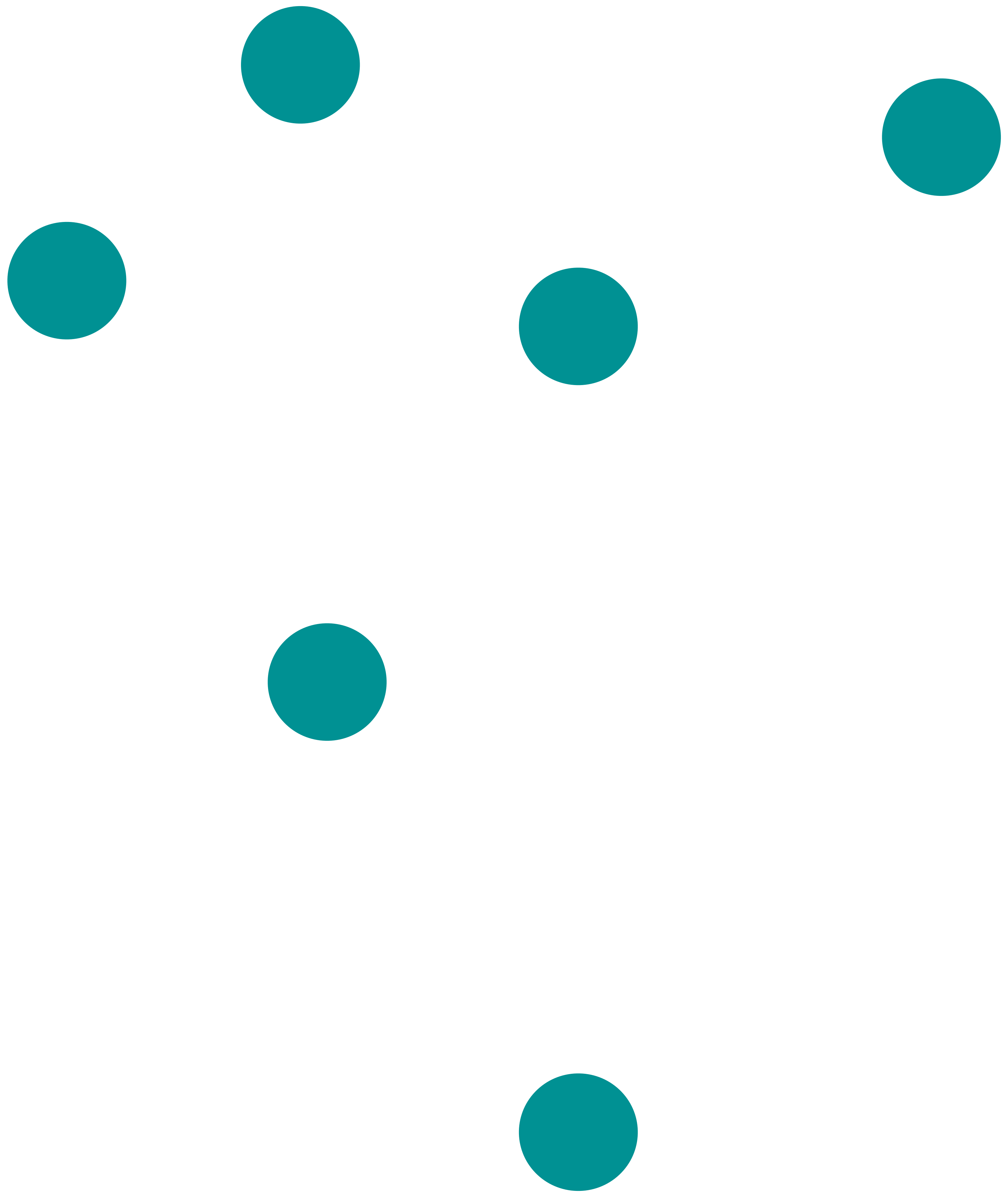
3. Find Eulerian Tour W

4. Traverse W once using shortcuts s.t. each vertex is visited exactly once

$\Rightarrow$ Hamiltonian Cycle C

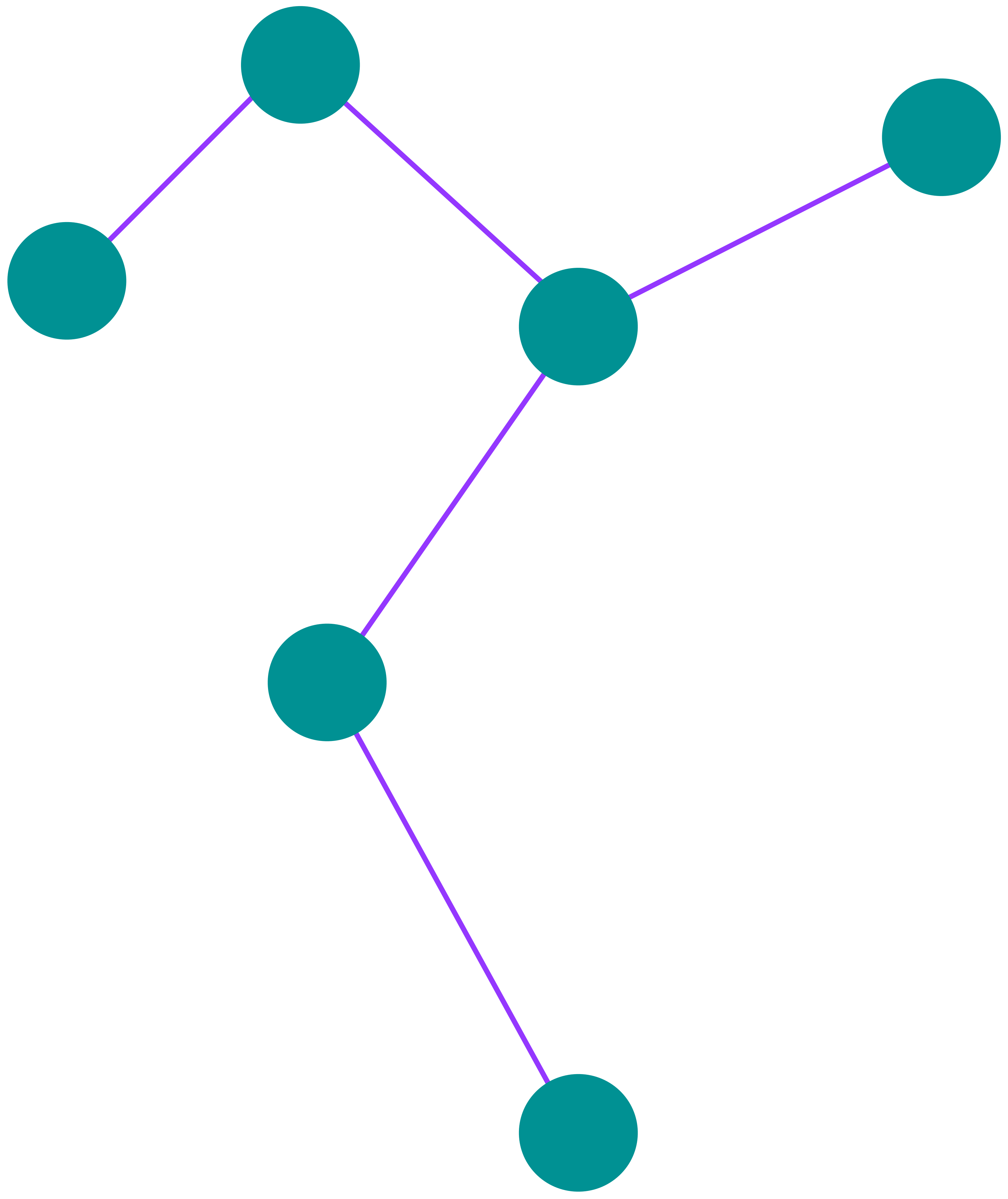
Metric TSP : 2-Approximation

Correctness



Metric TSP : 2-Approximation

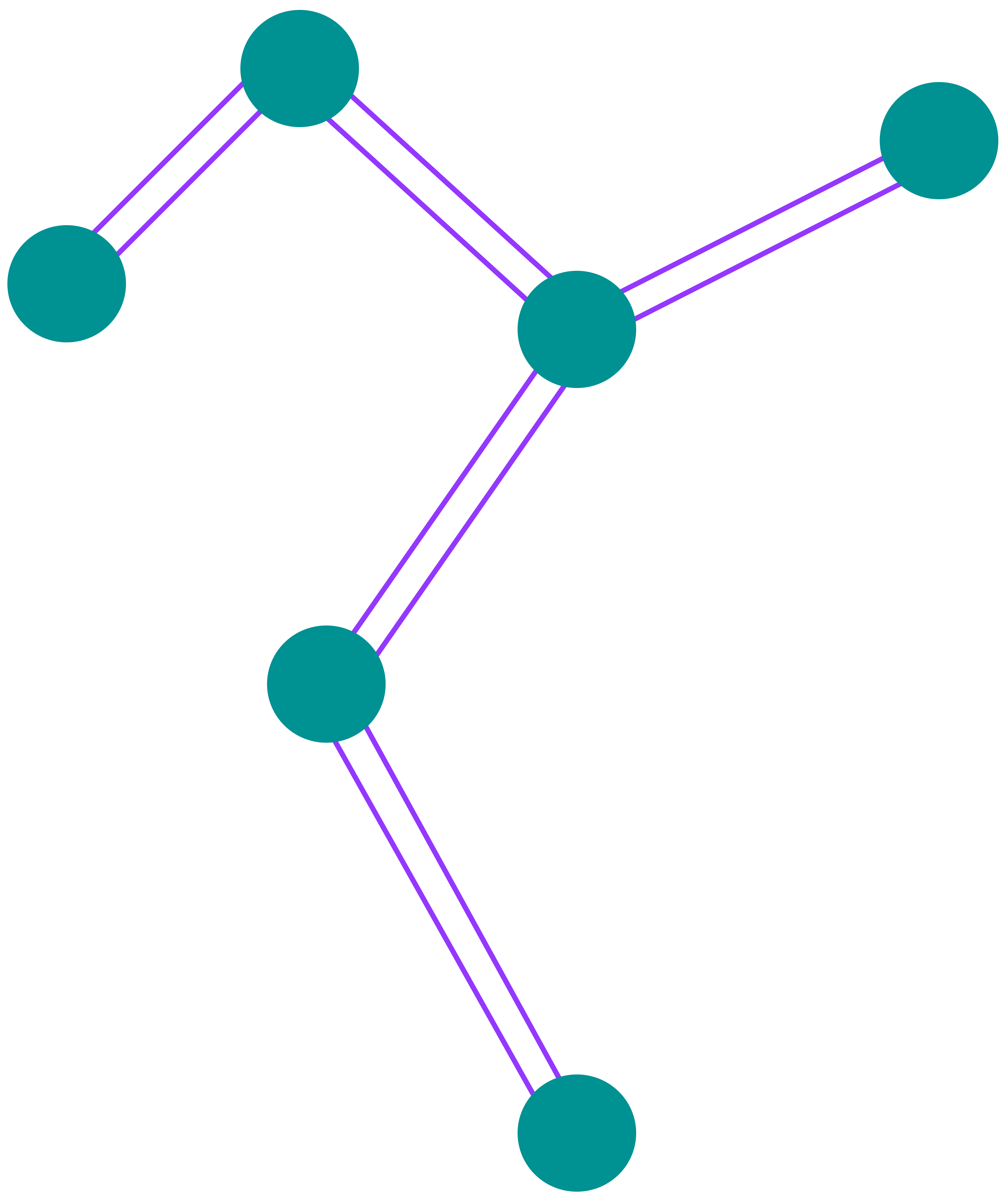
Correctness



1. Find the MST T $\quad l(T) \leq OPT(K_n, l)$

Metric TSP : 2-Approximation

Correctness

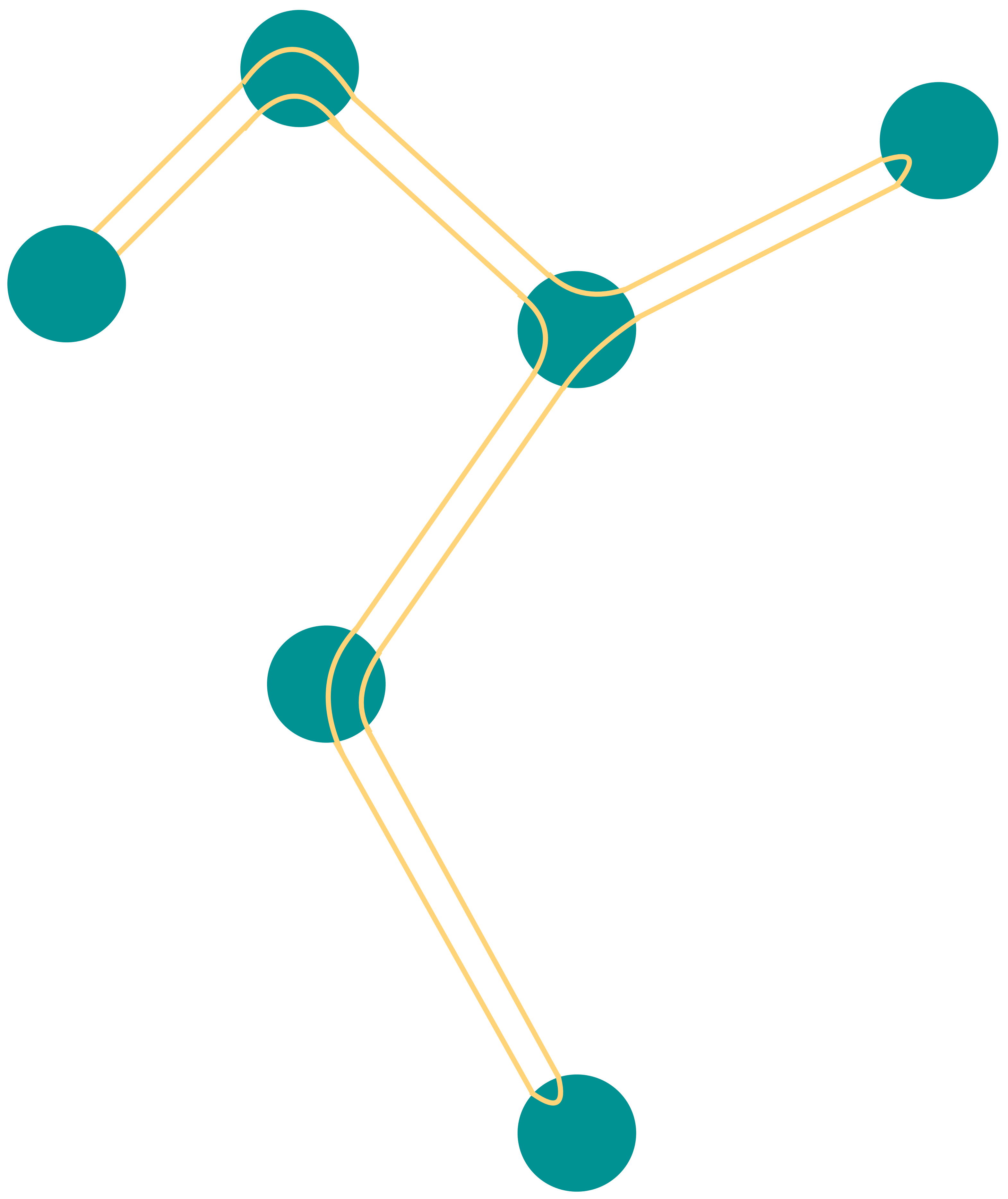


1. Find the MST T $\quad l(T) \leq OPT(K_n, l)$

2. Duplicate all edges of T $\quad 2\,l(T) \leq 2\,OPT(K_n, l)$

Metric TSP : 2-Approximation

Correctness



1. Find the MST T $\quad l(T) \leq OPT(K_n, l)$

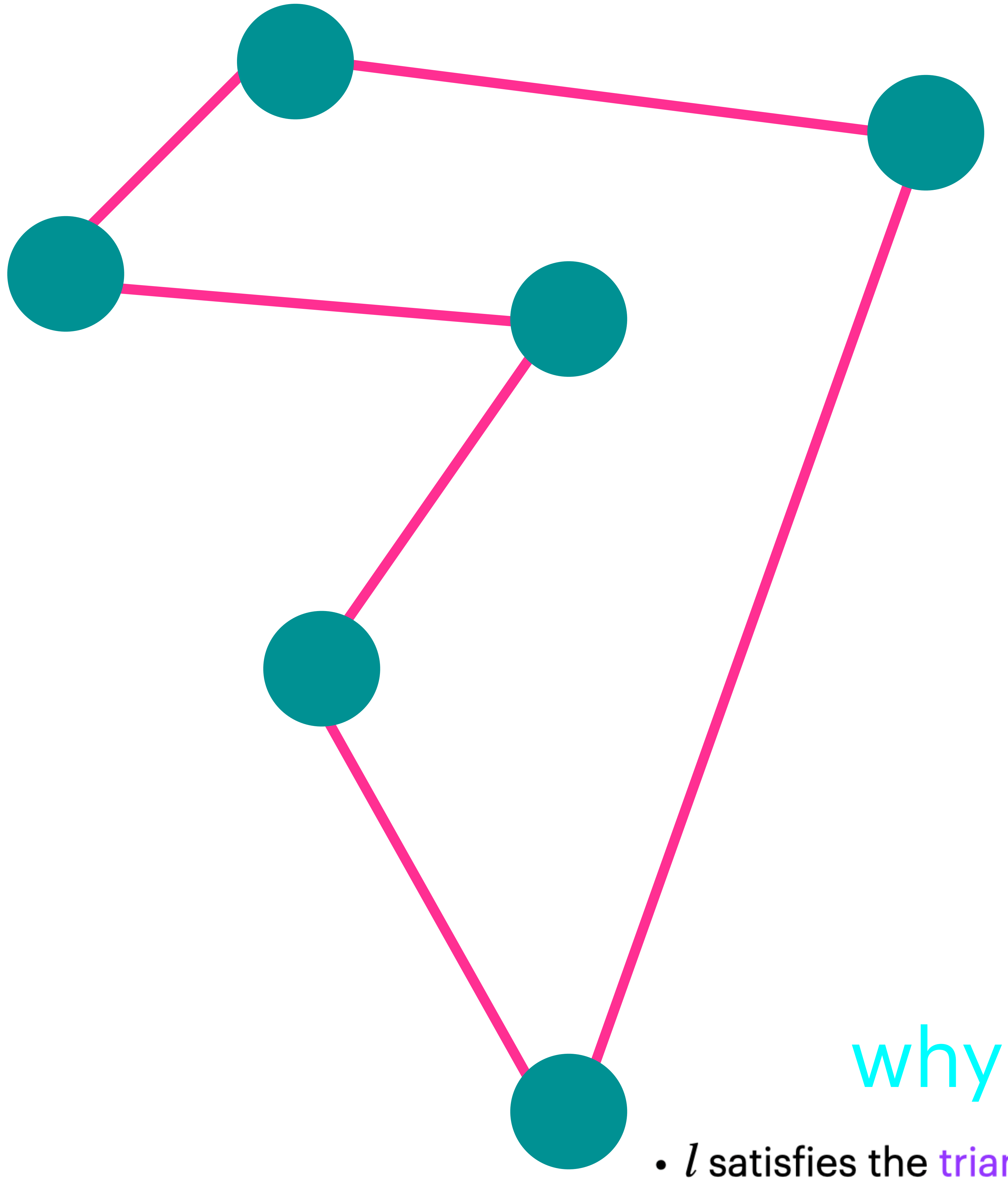
2. Duplicate all edges of T $\quad 2\,l(T) \leq 2\,OPT(K_n, l)$

3. Find Eulerian Tour W

$$l(W) = 2\,l(T) \leq 2\,OPT(K_n, l)$$

Metric TSP : 2-Approximation

Correctness



1. Find the MST T $\qquad l(T) \leq OPT(K_n, l)$

2. Duplicate all edges of T $\qquad 2\, l(T) \leq 2\, OPT(K_n, l)$

3. Find Eulerian Tour W

$$l(W) = 2\, l(T) \leq 2\, OPT(K_n, l)$$

4. Traverse W once using shortcuts

s.t. each vertex is visited exactly once $\quad \Rightarrow$ Hamiltonian Cycle C

why ?

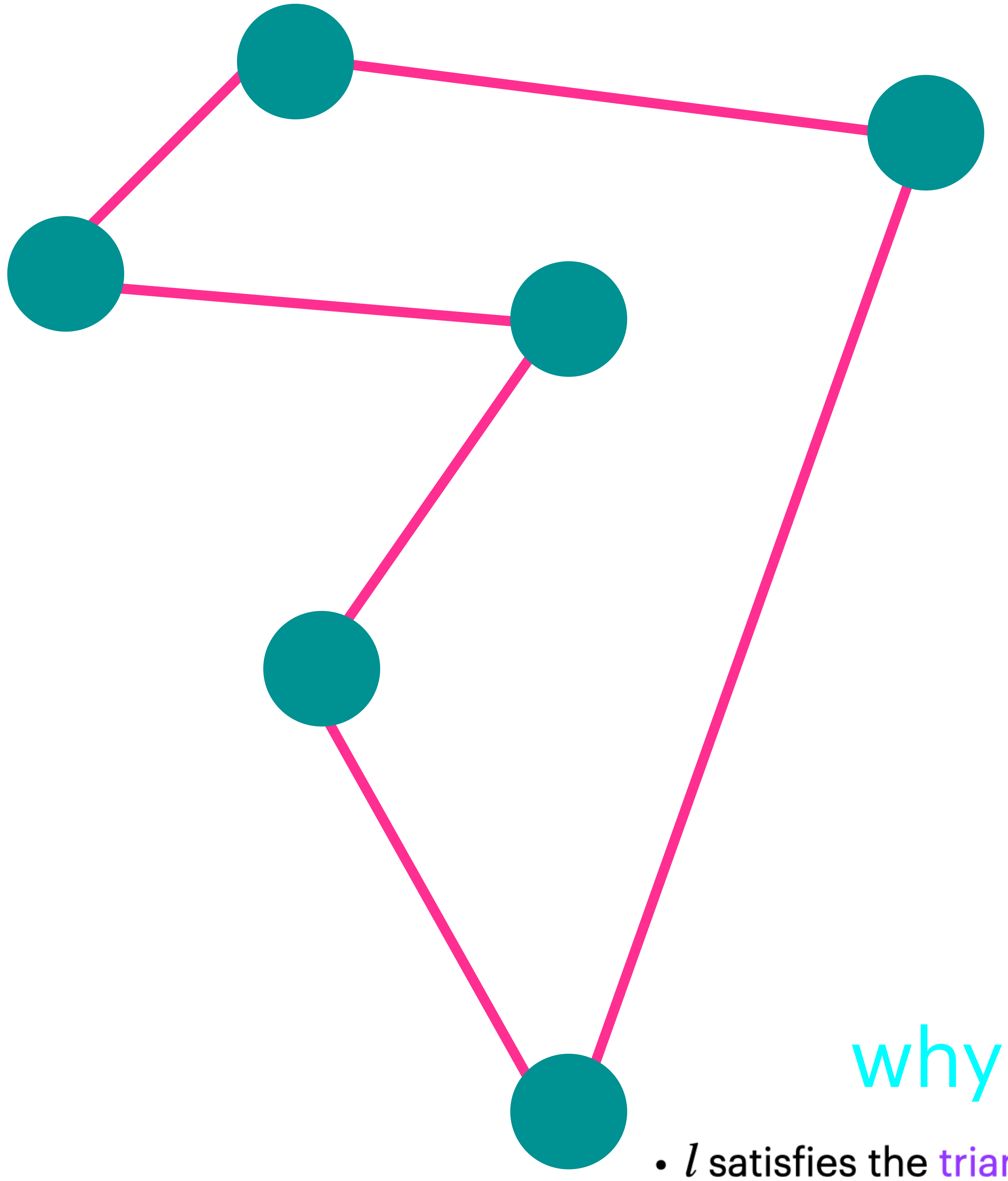
- l satisfies the triangle inequality

$l(x,z) \leq l(x,y) + l(y,z)$

$$l(C) \leq l(W) = 2\, l(T) \leq 2\, OPT(K_n, l)$$

Metric TSP : 2-Approximation

Correctness



1. Find the MST T $\quad l(T) \leq OPT(K_n, l)$

2. Duplicate all edges of T $\quad 2\,l(T) \leq 2\,OPT(K_n, l)$

3. Find Eulerian Tour W

$$l(W) = 2\,l(T) \leq 2\,OPT(K_n, l)$$

4. Traverse W once using shortcuts

s.t. each vertex is visited exactly once $\quad \Rightarrow$ Hamiltonian Cycle C

why ?

- l satisfies the triangle inequality

$l(x,z) \leq l(x,y) + l(y,z)$

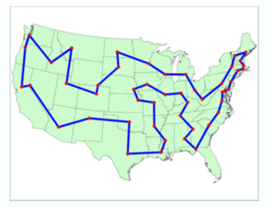
$$l(C) \leq l(W) = 2\,l(T) \leq 2\,OPT(K_n, l)$$

Metric TSP : 2-Approximation

Correctness

Goal : **Metric TSP : 2-Approximation**

Problem Description

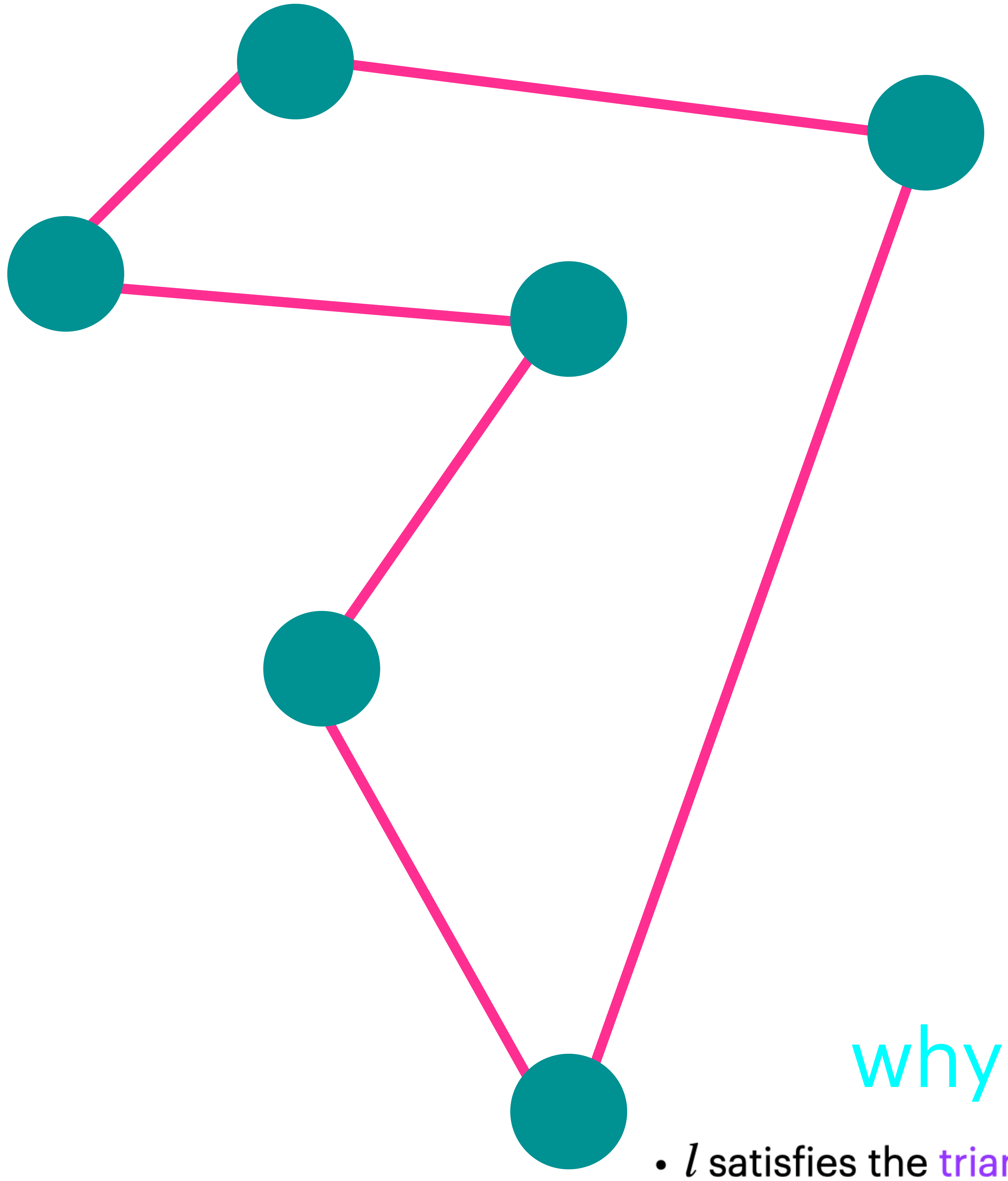


Given :
- A complete Graph K_n of n vertices
- Distances l inbetween every 2 vertex $l : \binom{[n]}{2} \to R$
- l satisfies the triangle inequality $l(x,z) \leq l(x,y) + l(y,z)$

To find :
- Hamiltonian Cycle C s.t.

$$l(C) \leq 2\, l(OPT)$$

where $OPT = \min_{H : \text{Hamiltonian Cycle}} \sum_{e \in E(H)} l(e)$



1. Find the MST T $\qquad l(T) \leq OPT(K_n, l)$

2. Duplicate all edges of T $\qquad 2\, l(T) \leq 2\, OPT(K_n, l)$

3. Find Eulerian Tour W

$$l(W) = 2\, l(T) \leq 2\, OPT(K_n, l)$$

4. Traverse W once using shortcuts

s.t. each vertex is visited exactly once $\Rightarrow$ Hamiltonian Cycle C

why ?

- l satisfies the triangle inequality

$l(x,z) \leq l(x,y) + l(y,z)$

$$l(C) \leq l(W) = 2\, l(T) \leq 2\, OPT(K_n, l)$$

Questions

Feedbacks , Recommendations

Nil Ozer