

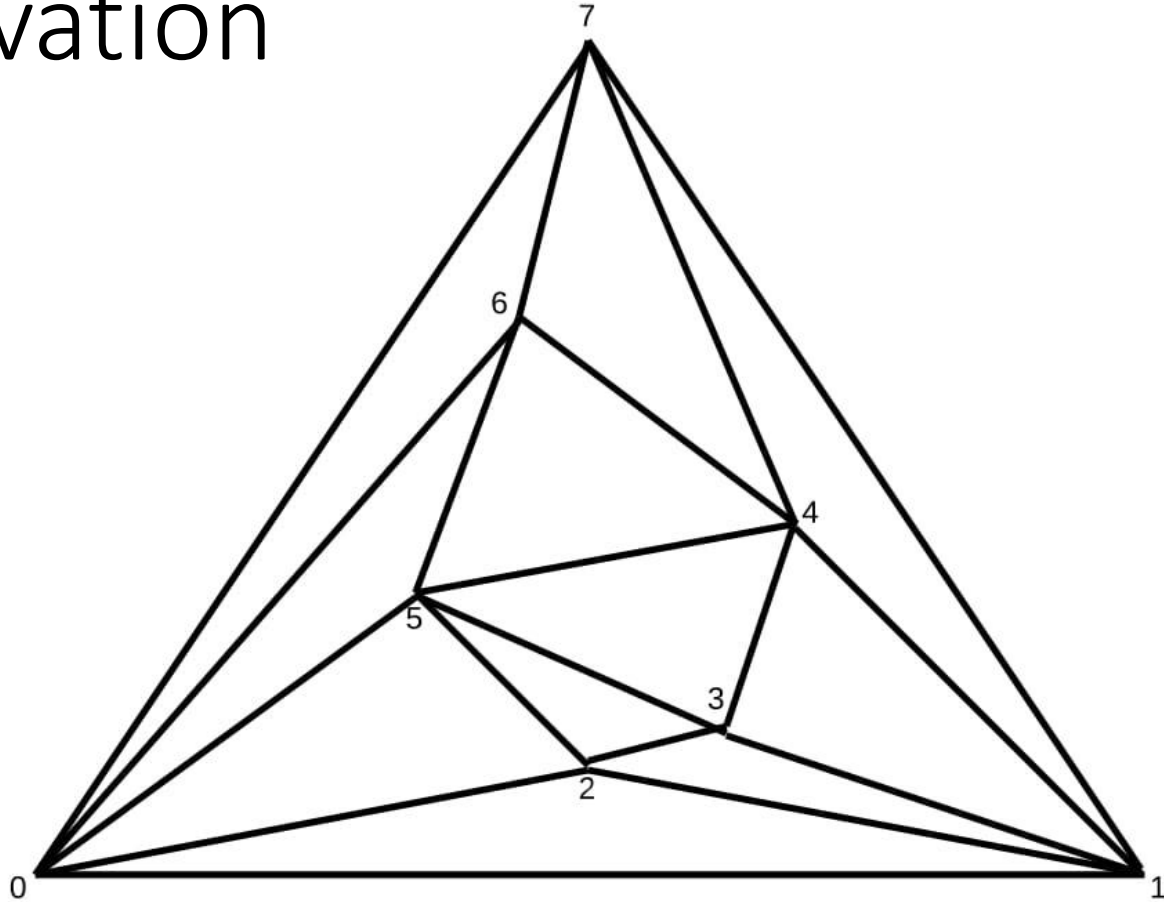
Succinct Representation of Labeled Graphs

Matthias Dürksen

Seminar on Algorithms for Compressed Graphs

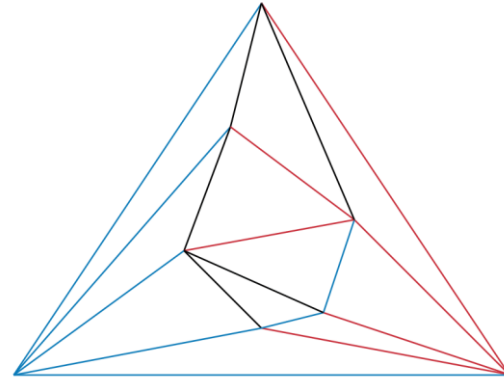
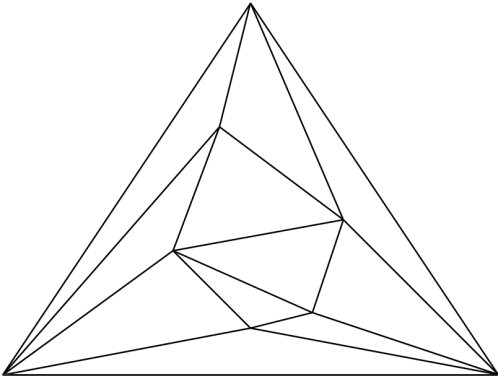
4th February 2019

Motivation



$G=(V,E)$?

Outline

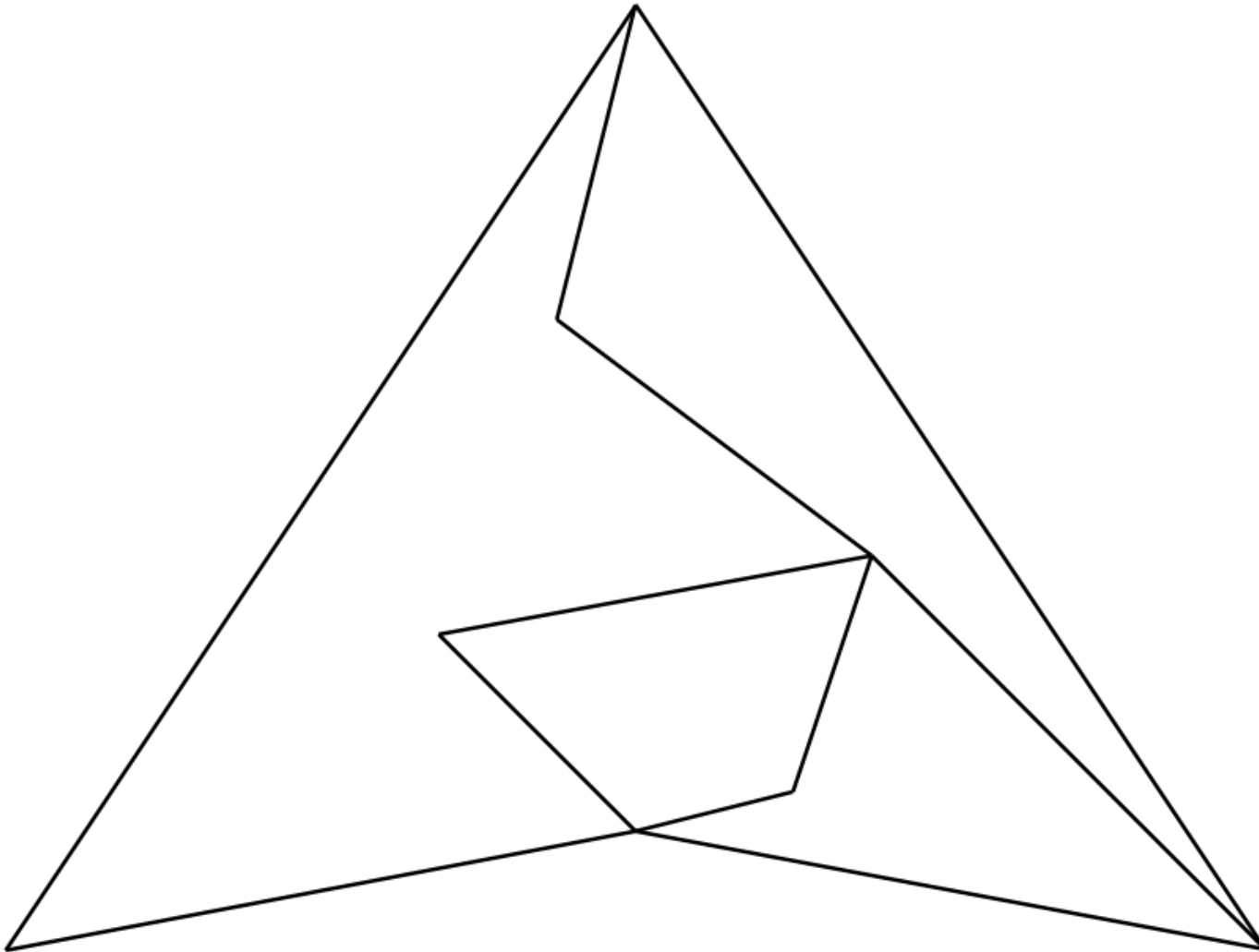

$$(0 \quad (0 \quad 0)) \quad (0 \quad 0 \quad 0)$$
$$\left[\begin{array}{cc} \left[\begin{array}{cc} \left[\begin{array}{c} 1 \\ 0 \end{array} \right] & \left[\begin{array}{c} 0 \\ 1 \end{array} \right] \end{array} \right] & \left[\begin{array}{c} 0 \\ 0 \end{array} \right] \end{array} \right]$$
$$\{ \{ \{ \} \} \{ \} \{ \} \}$$

The diagram illustrates the iterative process of building a valid parenthesis string. It starts with seven opening brackets '(', indexed 0 through 6. At each step, a closing bracket ')' is added to the end of the current sequence if it does not result in a prefix with more closing than opening brackets. The steps are as follows:

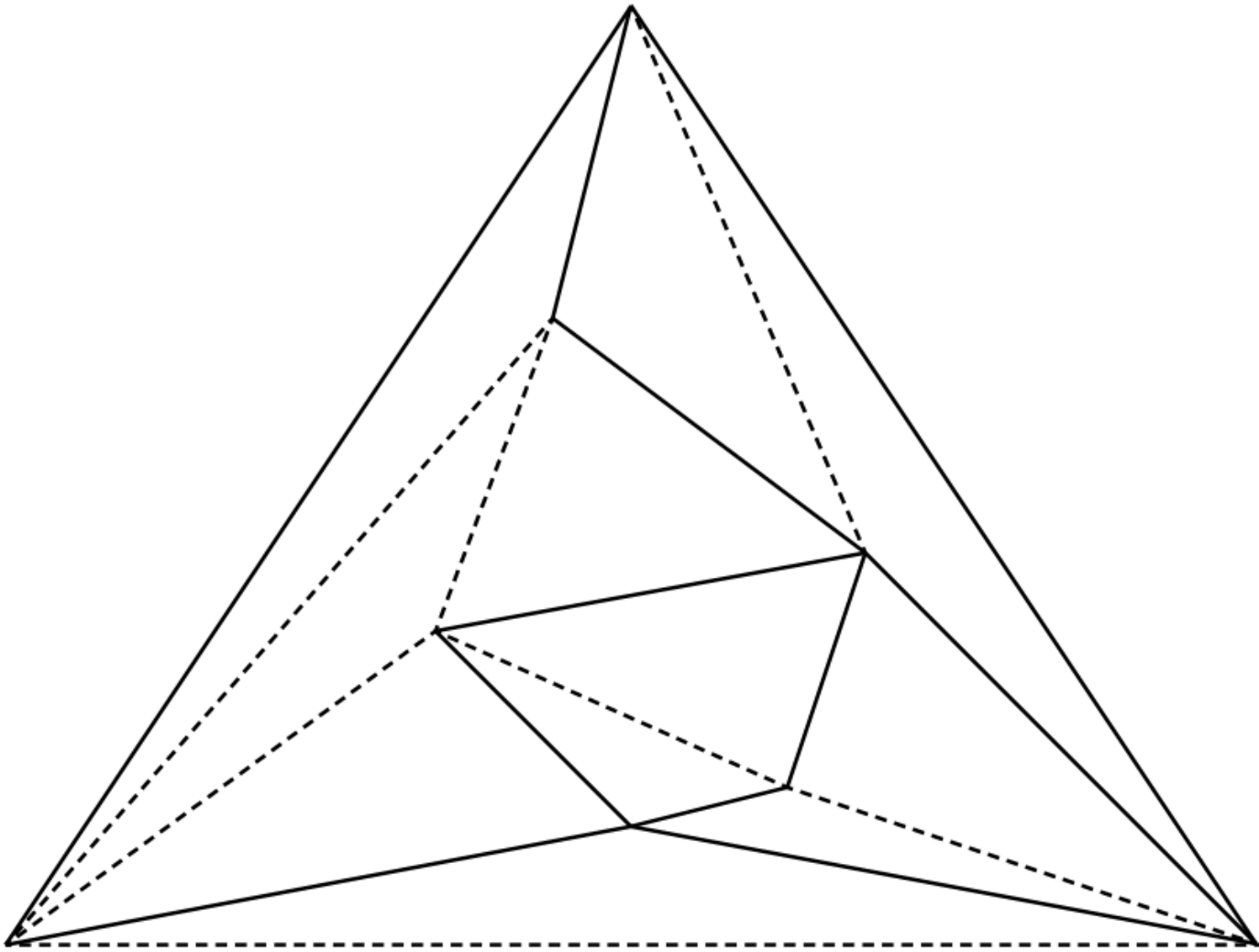
- Step 0:** Initial sequence: (
- Step 1:** Add '(' → ((
- Step 2:** Add '(' → (((
- Step 3:** Add ')' → ((()
- Step 4:** Add '(' → ((((
- Step 5:** Add ')' → (((())
- Step 6:** Add ')' → (((()))
- Step 7:** Add ')' → (((())))

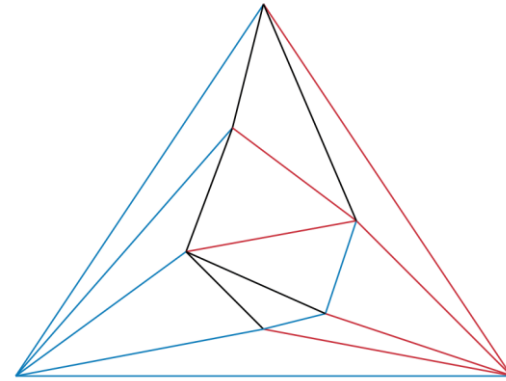
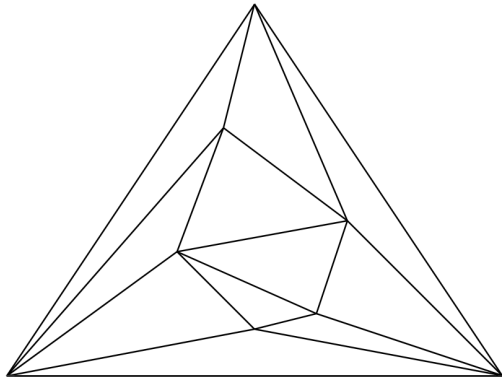
The final sequence after 7 steps is "((()))", which is a valid parenthesis string.

Planar Graphs

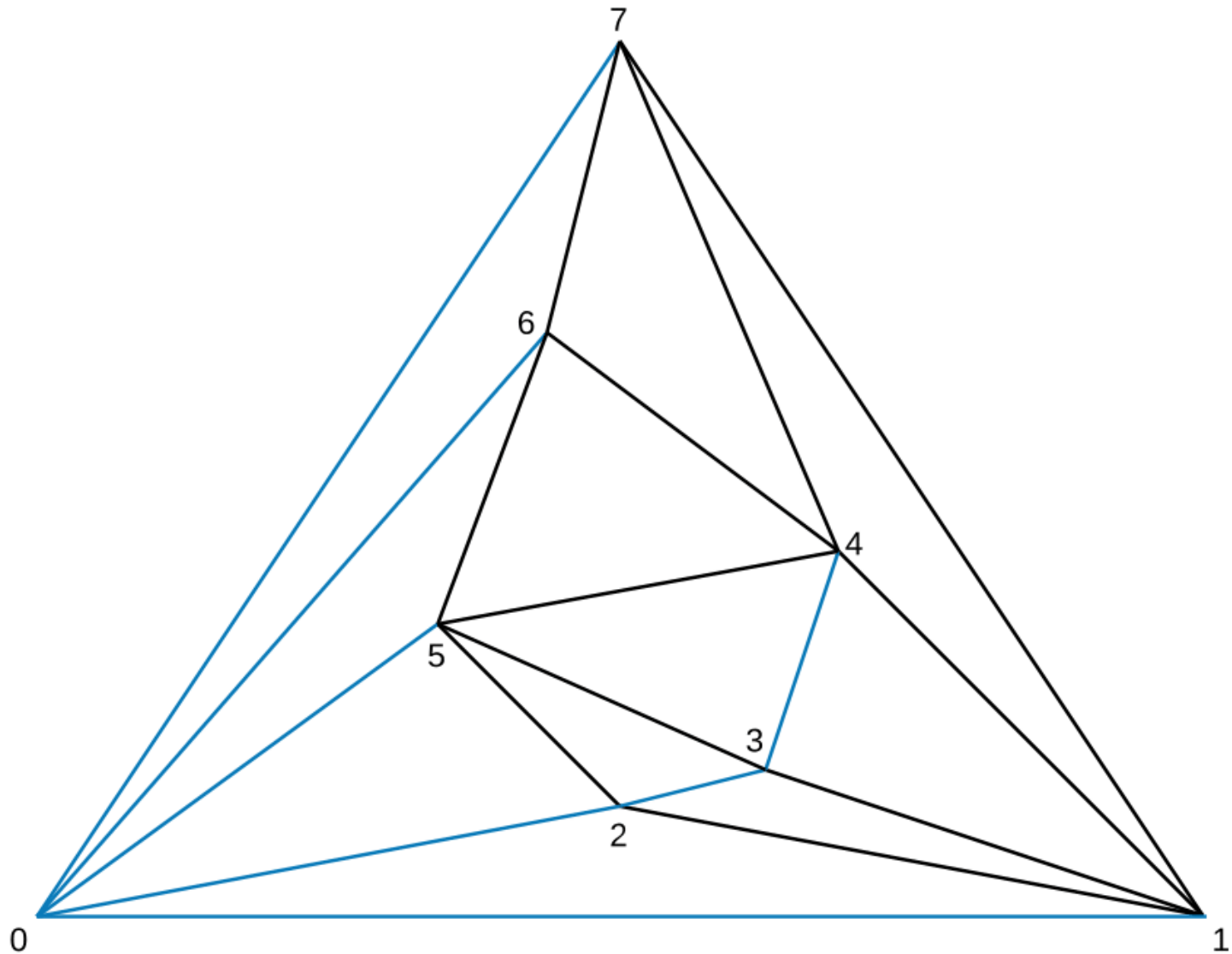


Triangulated Graphs

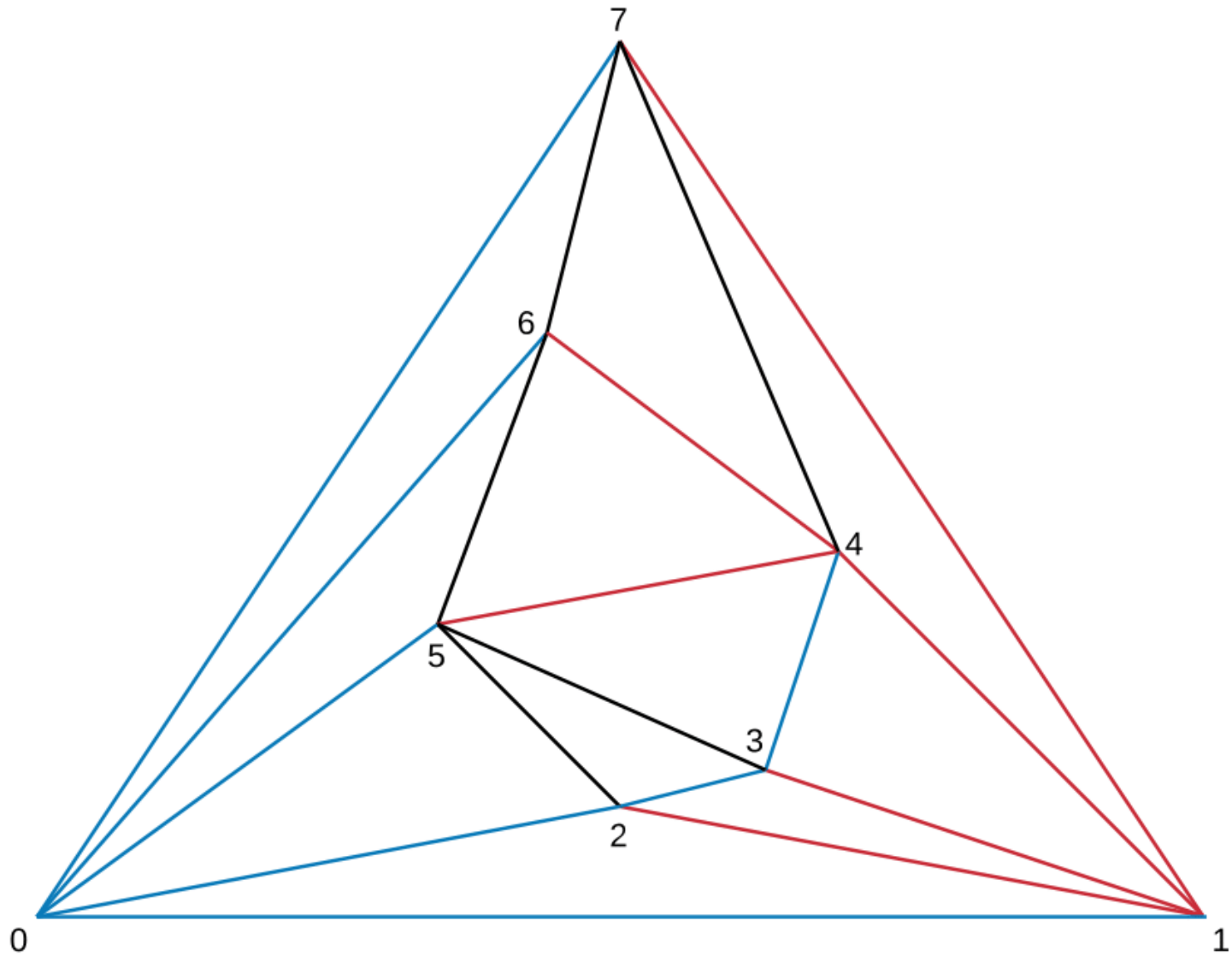




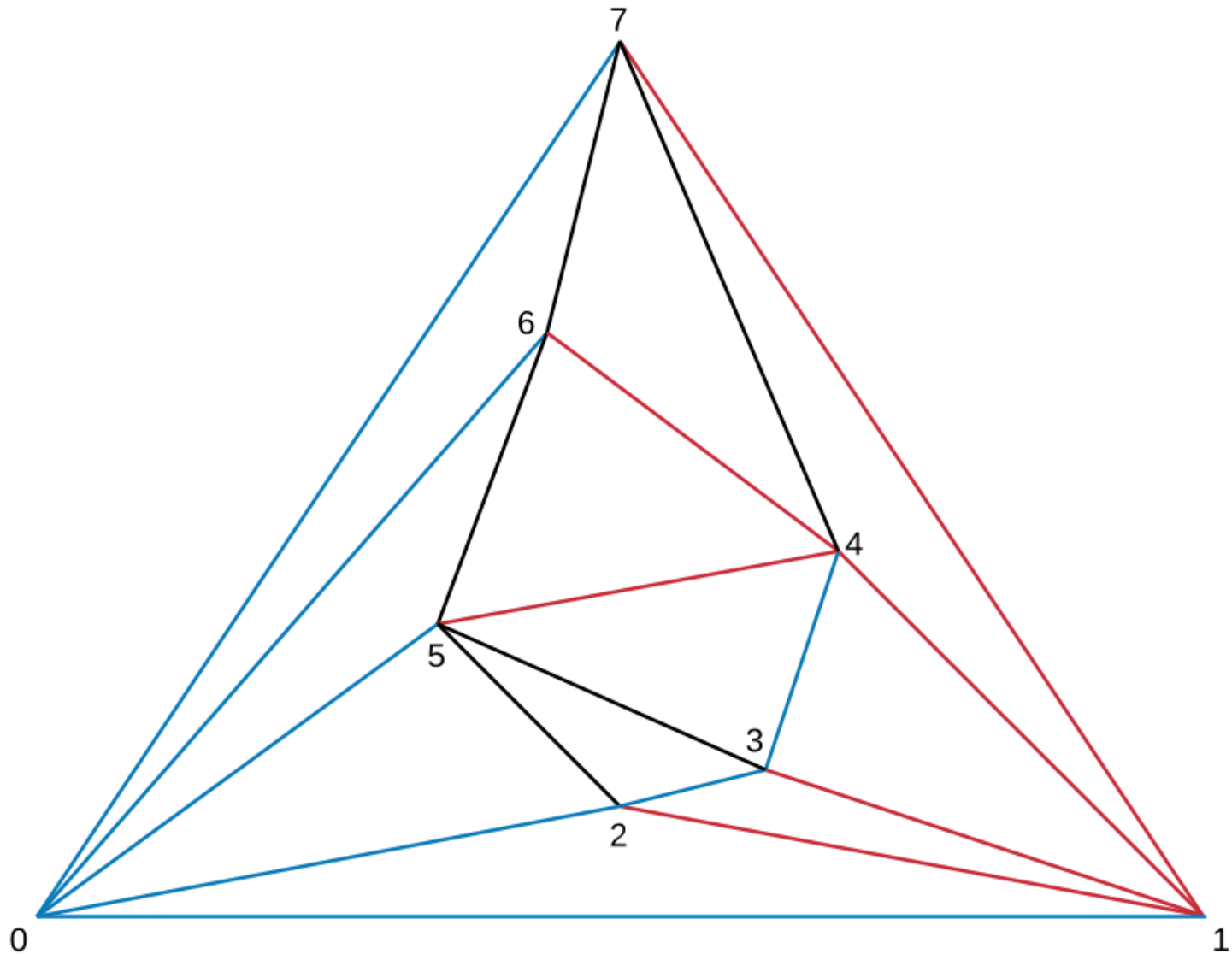
First Tree – Canonical Spanning Tree

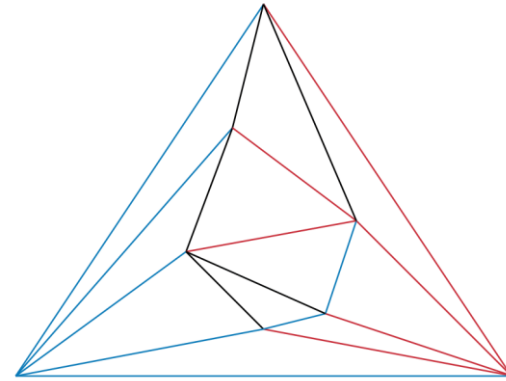
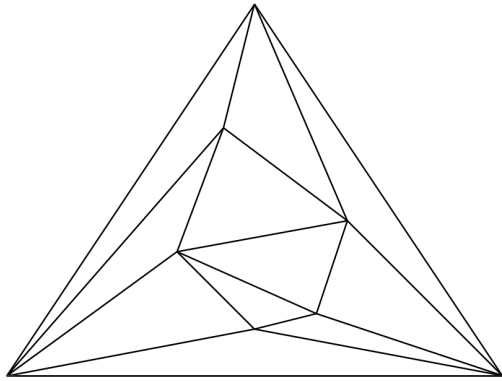


Second Tree



Third Tree



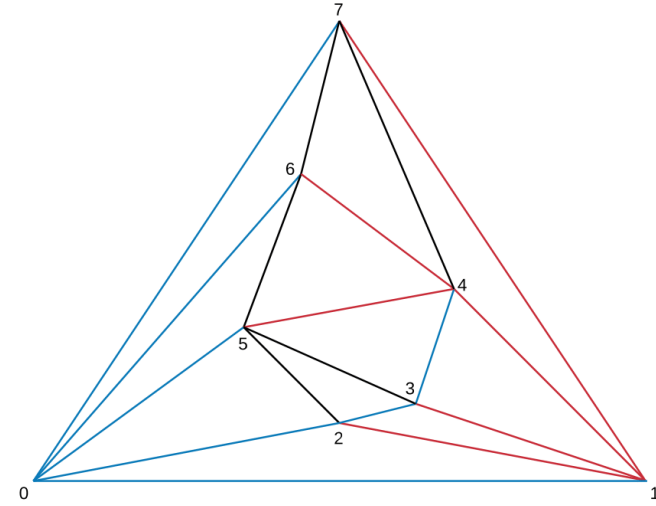
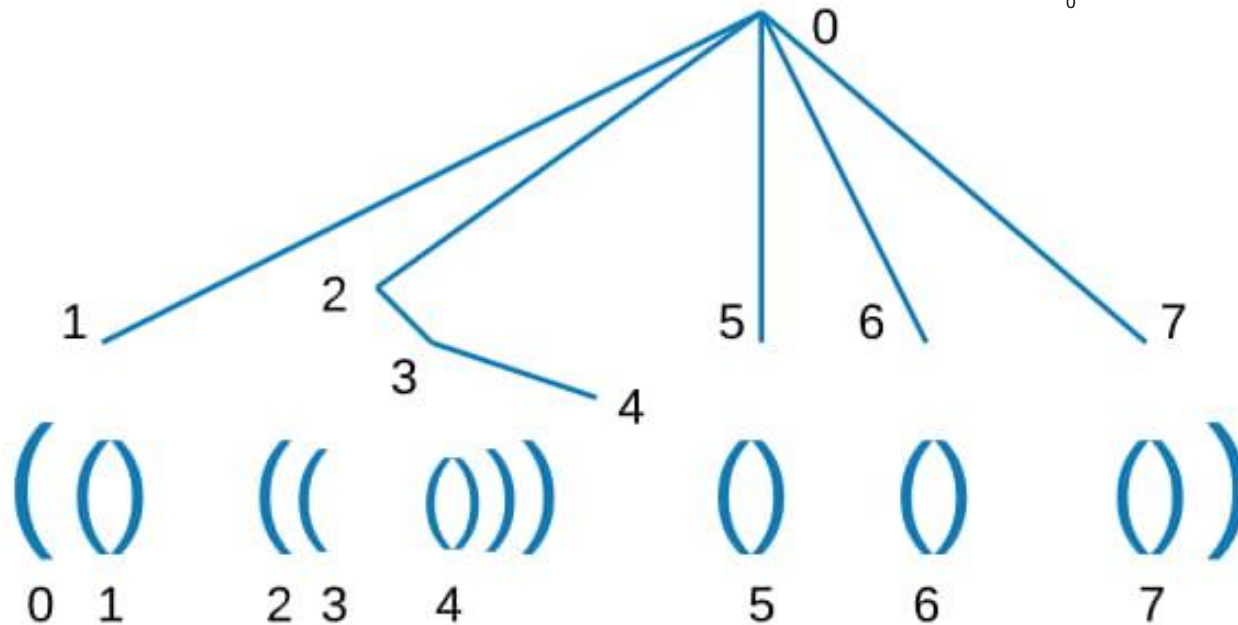


(0 ((0)) 0 0 0)

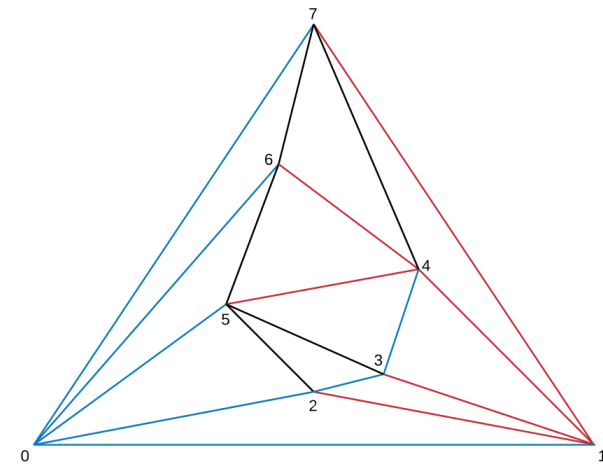
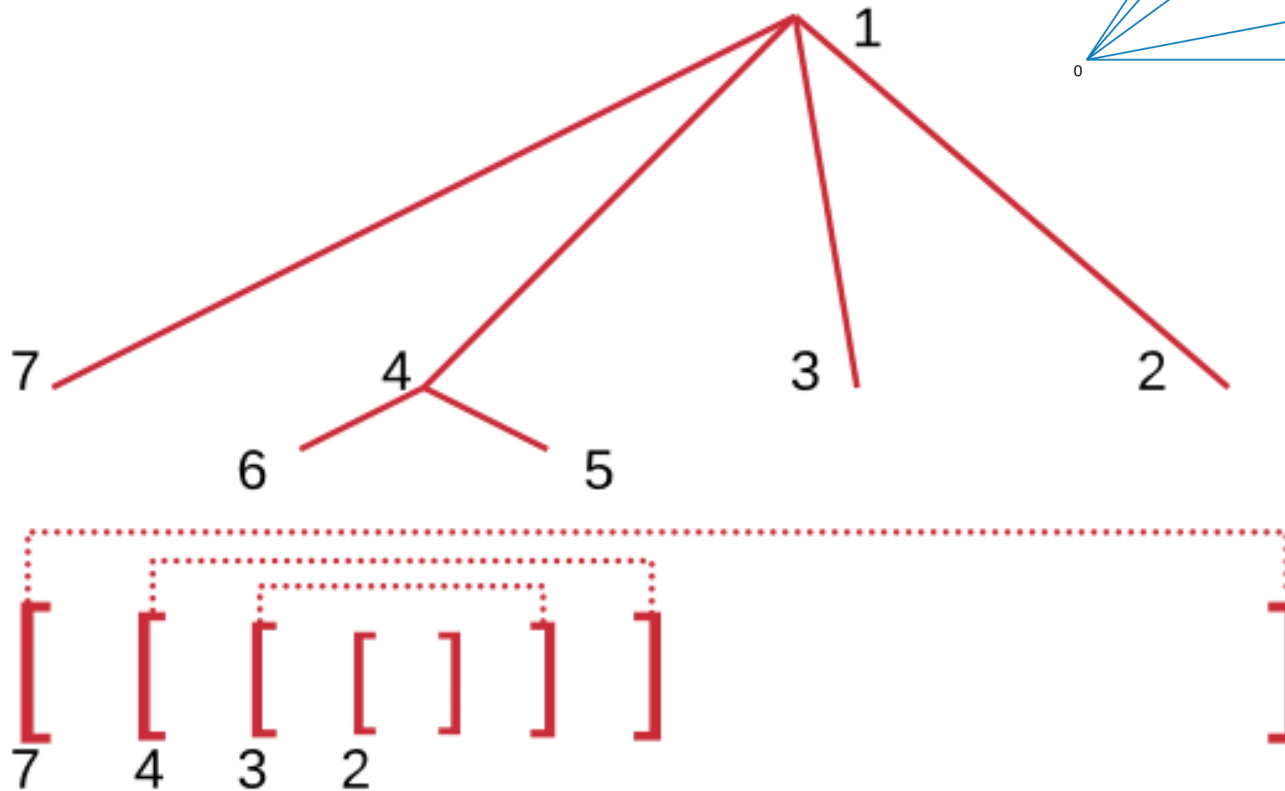
[[[[]]] [[]]]

{ { { } } { } { } }

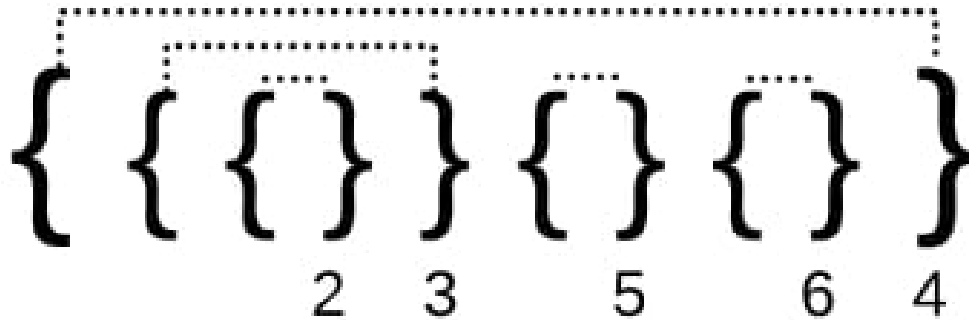
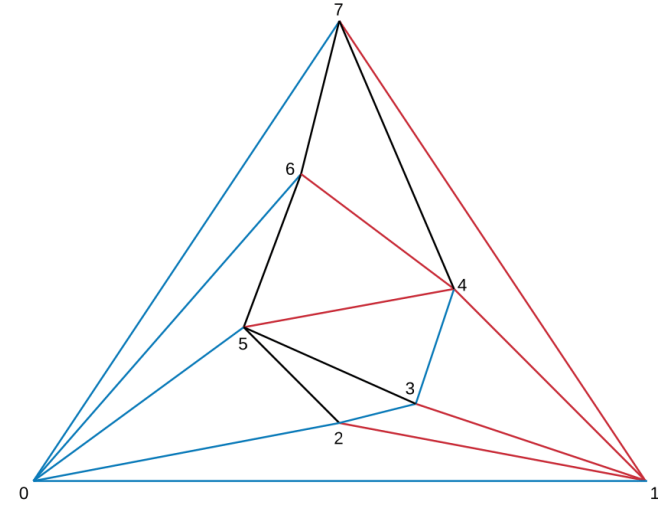
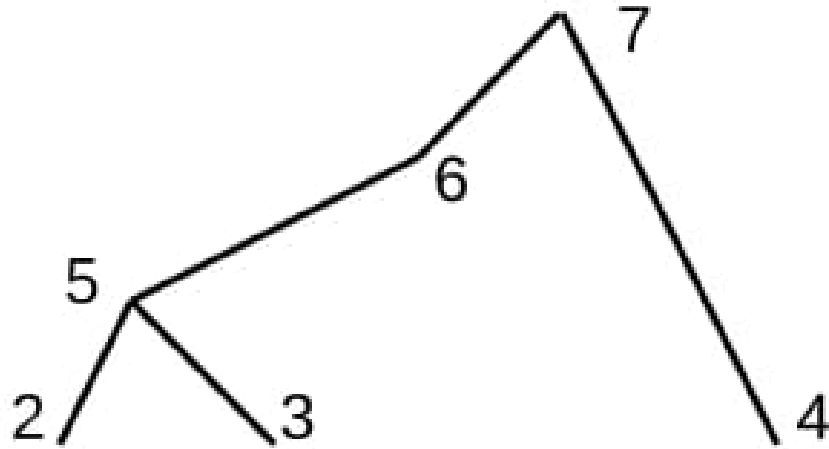
Parenthesis

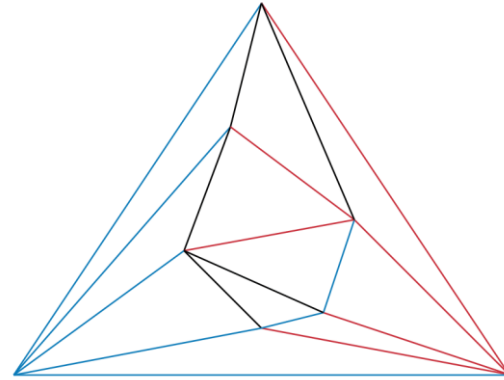


Parenthesis



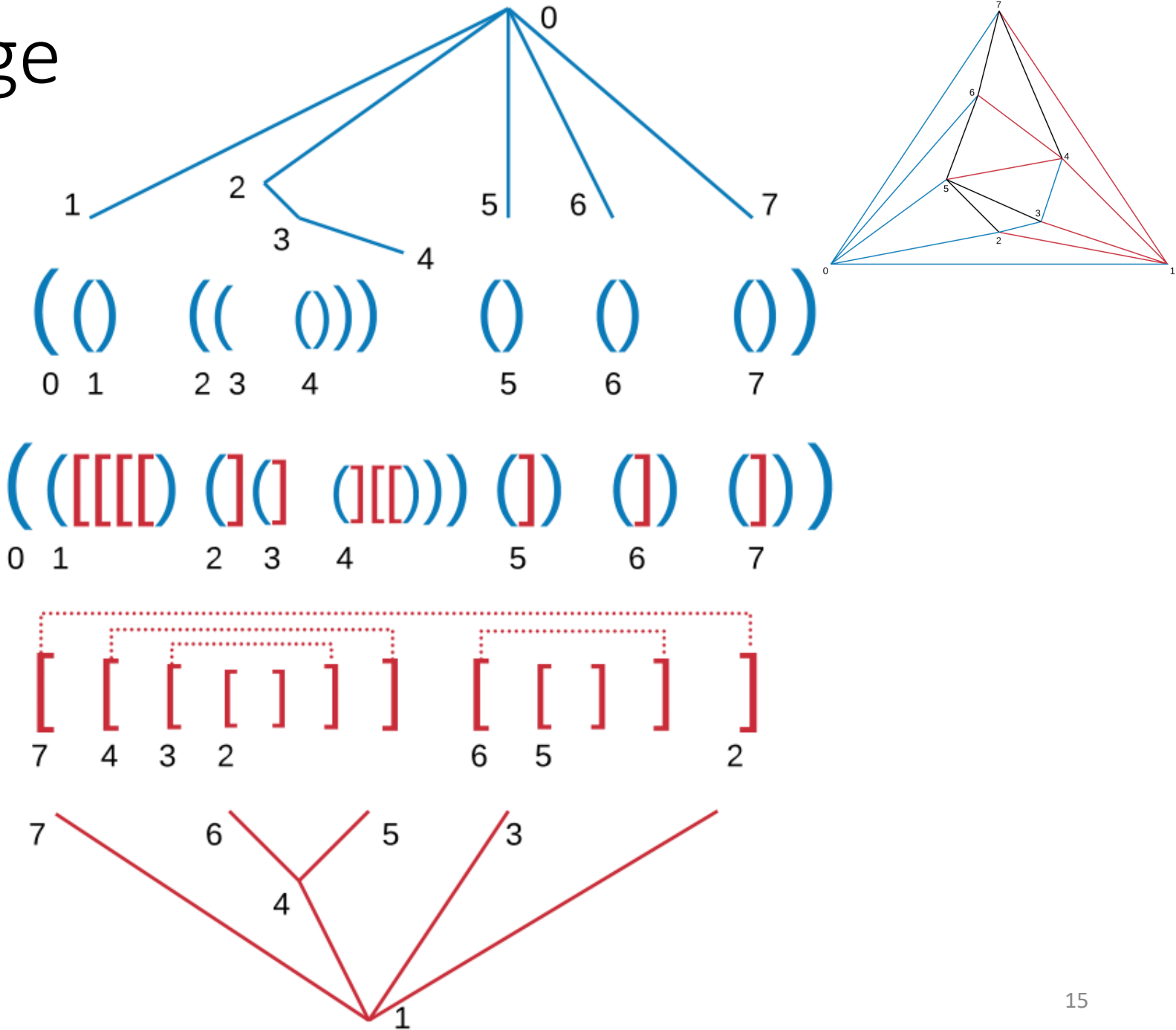
Parenthesis



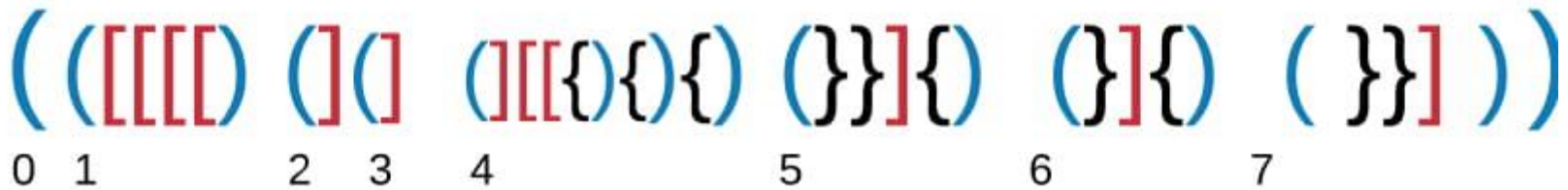
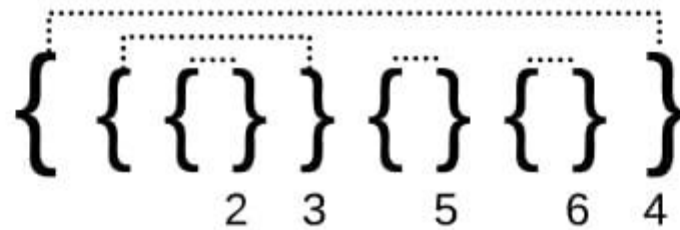
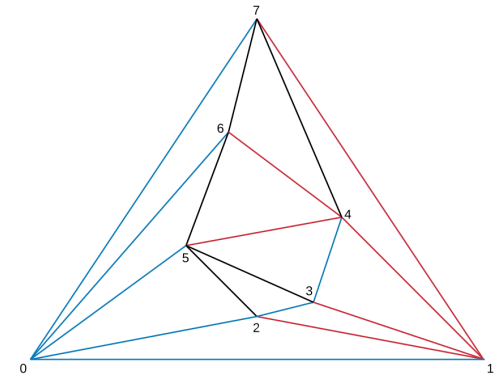
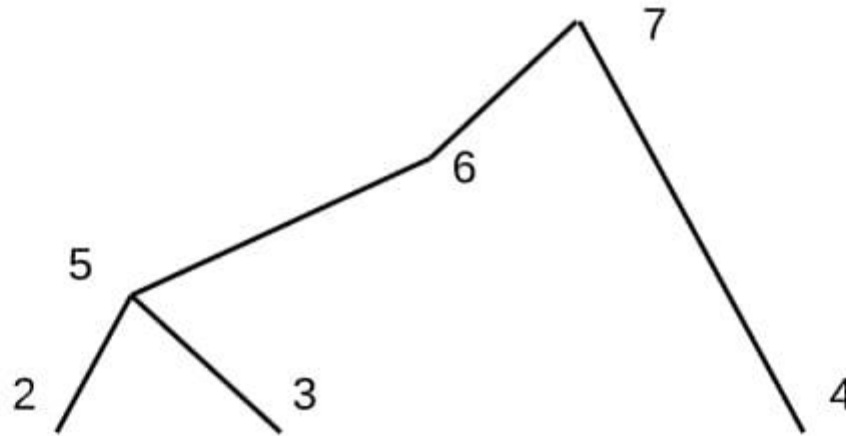

$$\{ \{ \{ \} \} \{ \} \{ \} \{ \} \}$$

0 1 2 3 4 5 6 7

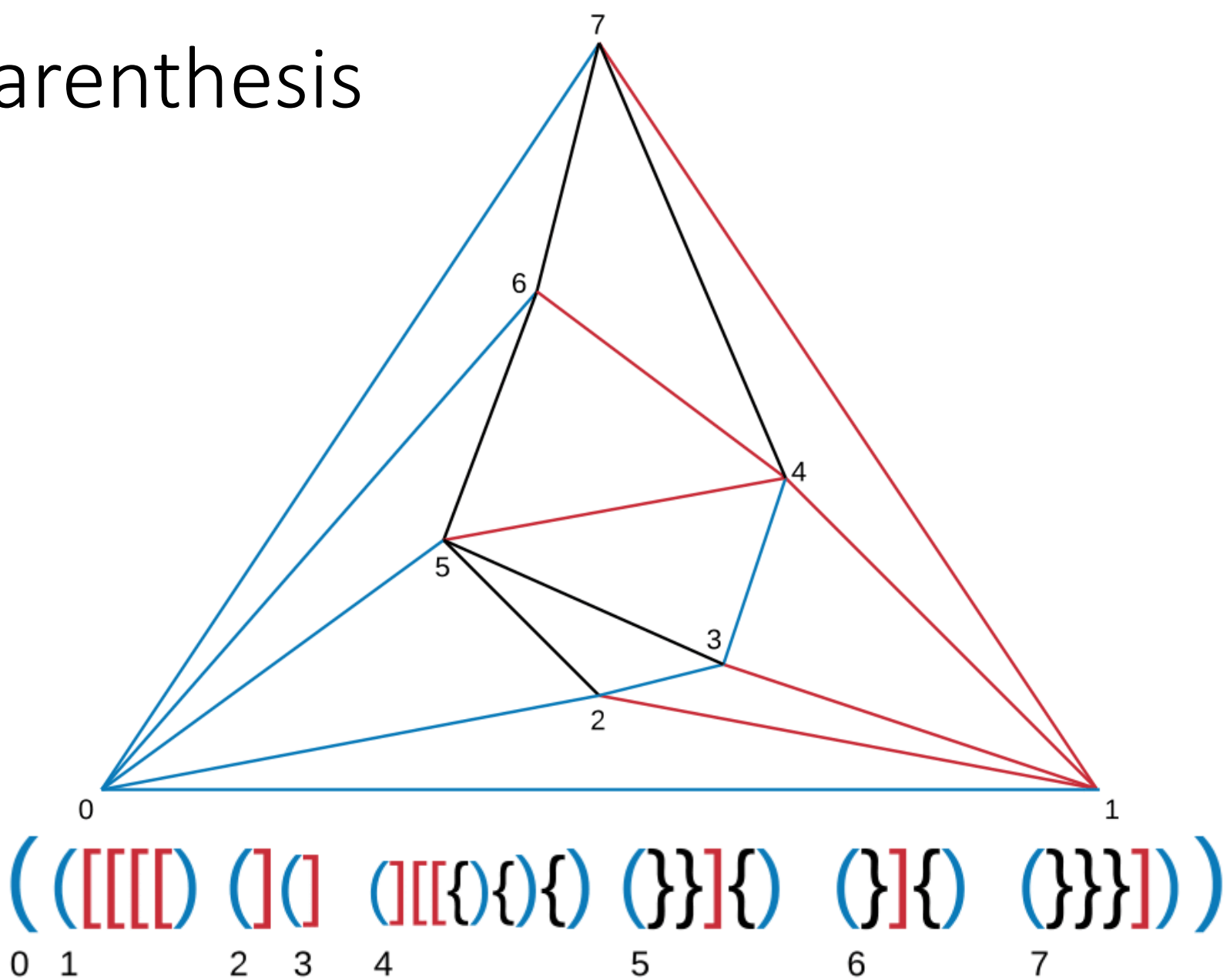
Merge



Merge



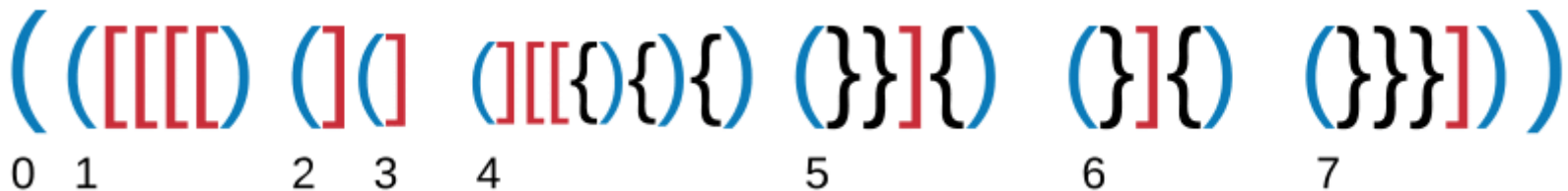
Parenthesis



Parenthesized representation

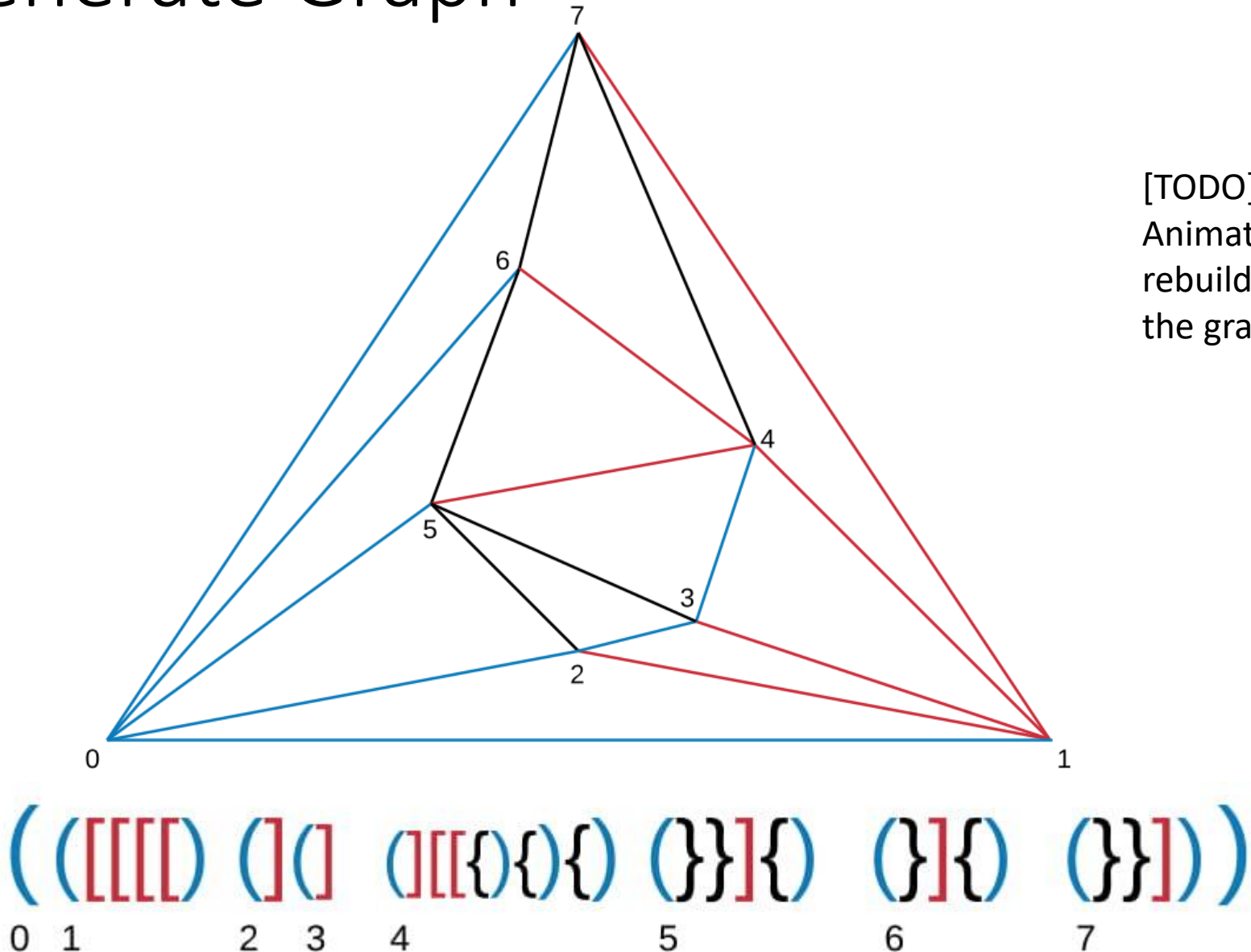
- Each type \rightarrow forms a correct parenthesis
- Two parenthesis per Edge

$\Rightarrow 2m \cdot 3$ Bits necessary



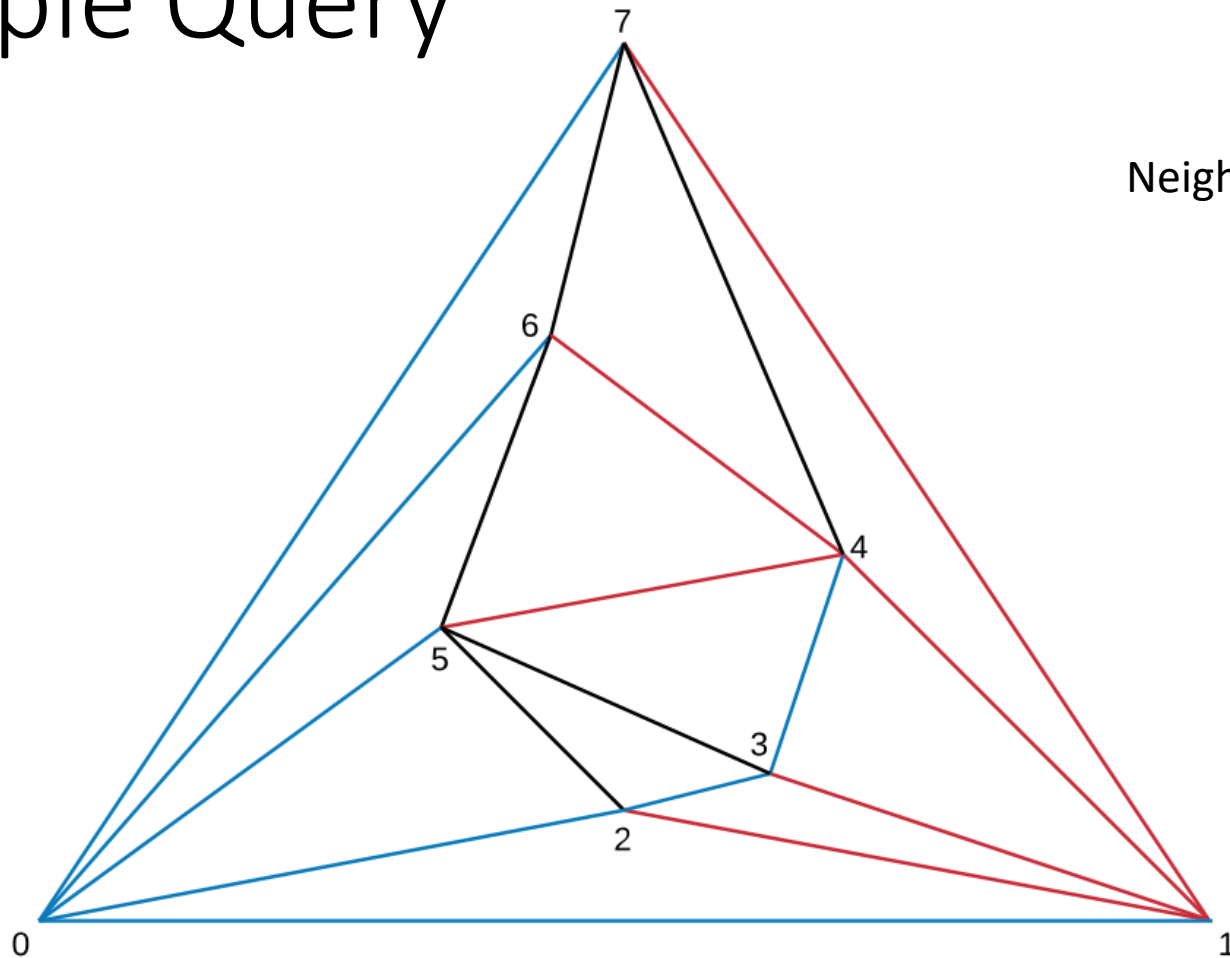
Generate Graph

[TODO]
Animated
rebuilding
the graph



Sample Query

Neighborhood Query



(([[[[]]]]) ([] ([] ([[[{ } { } { }]]] { }) ({ }] { } ({ } { }]))

0 1 2 3 4 5 6 7

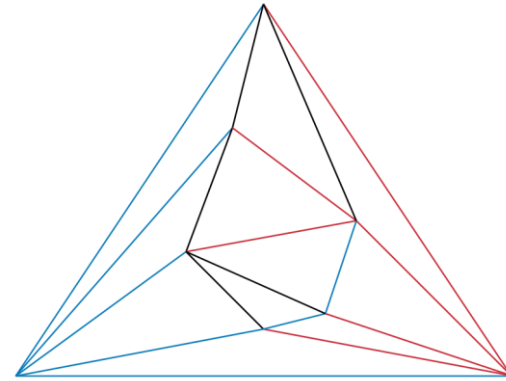
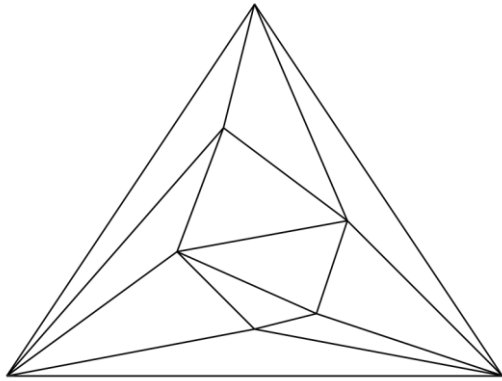
Parenthesized representation

- Why to use the parentheses
- Explain Results and supported functions

Extension

- Labeled Graphs: As always with a mapping from node id to Label
- [Some more words to it, depending on how much time remains]

Summary



(() (()) 0 0 0)

[[[[]]] [[]]]

{ { { } } { } { } }

(([[[[]]) ([]]) ([[[[]]) ([]]) ([]]) ([]]) ([]]) ([]]))

0 1 2 3 4 5 6 7

⇒ 6m Bits necessary

⇒ Query runtimes like uncompressed

End

Thank you for your attention!