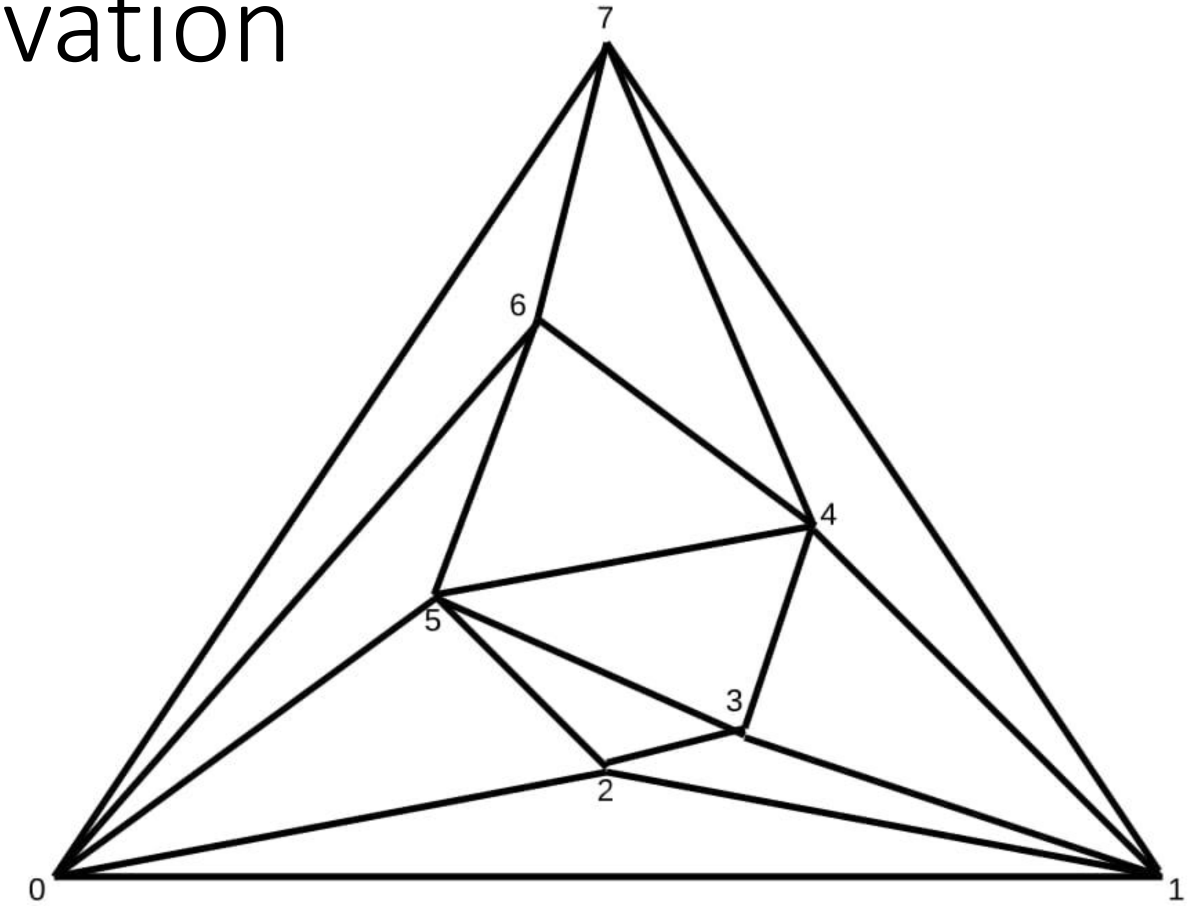
Succinct Representation of Labeled Graphs

Matthias Dürksen

Seminar on Algorithms for Compressed Graphs

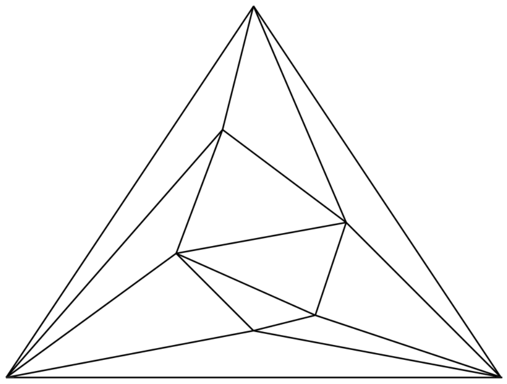
4th February 2019

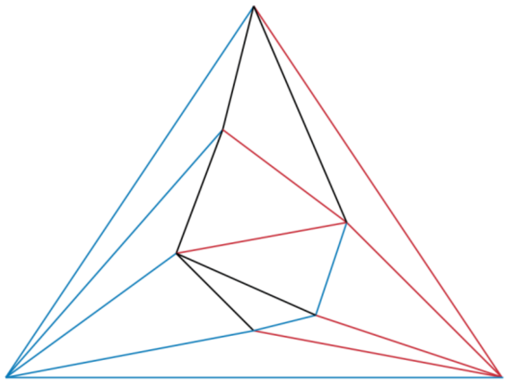
Motivation

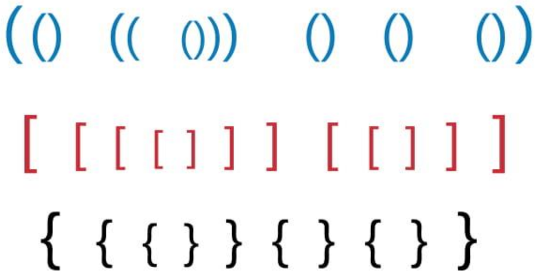


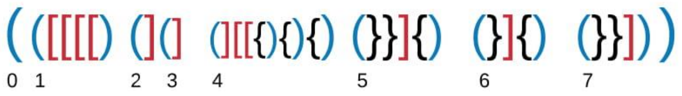
G=(V,E) ?

Outline

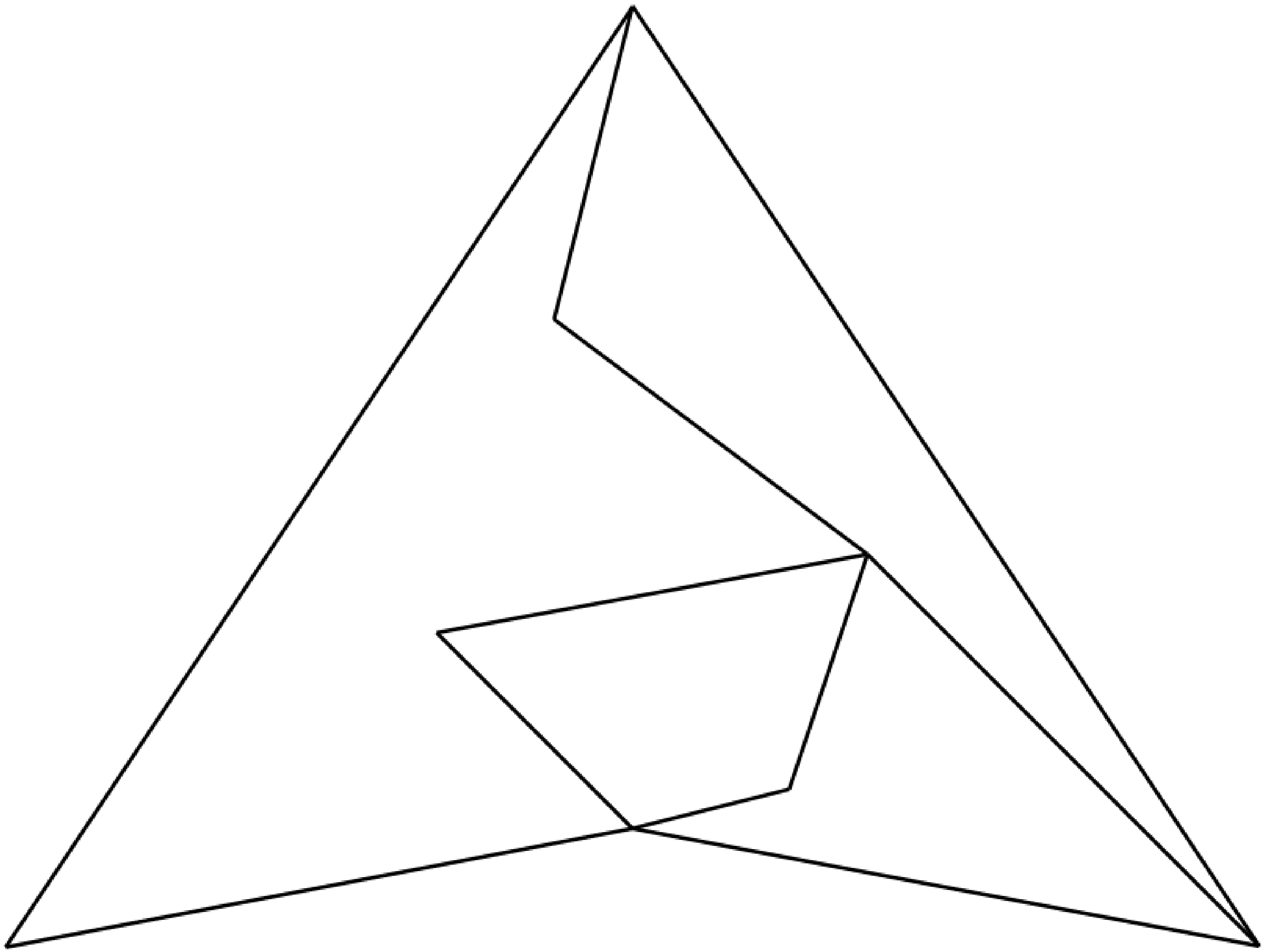




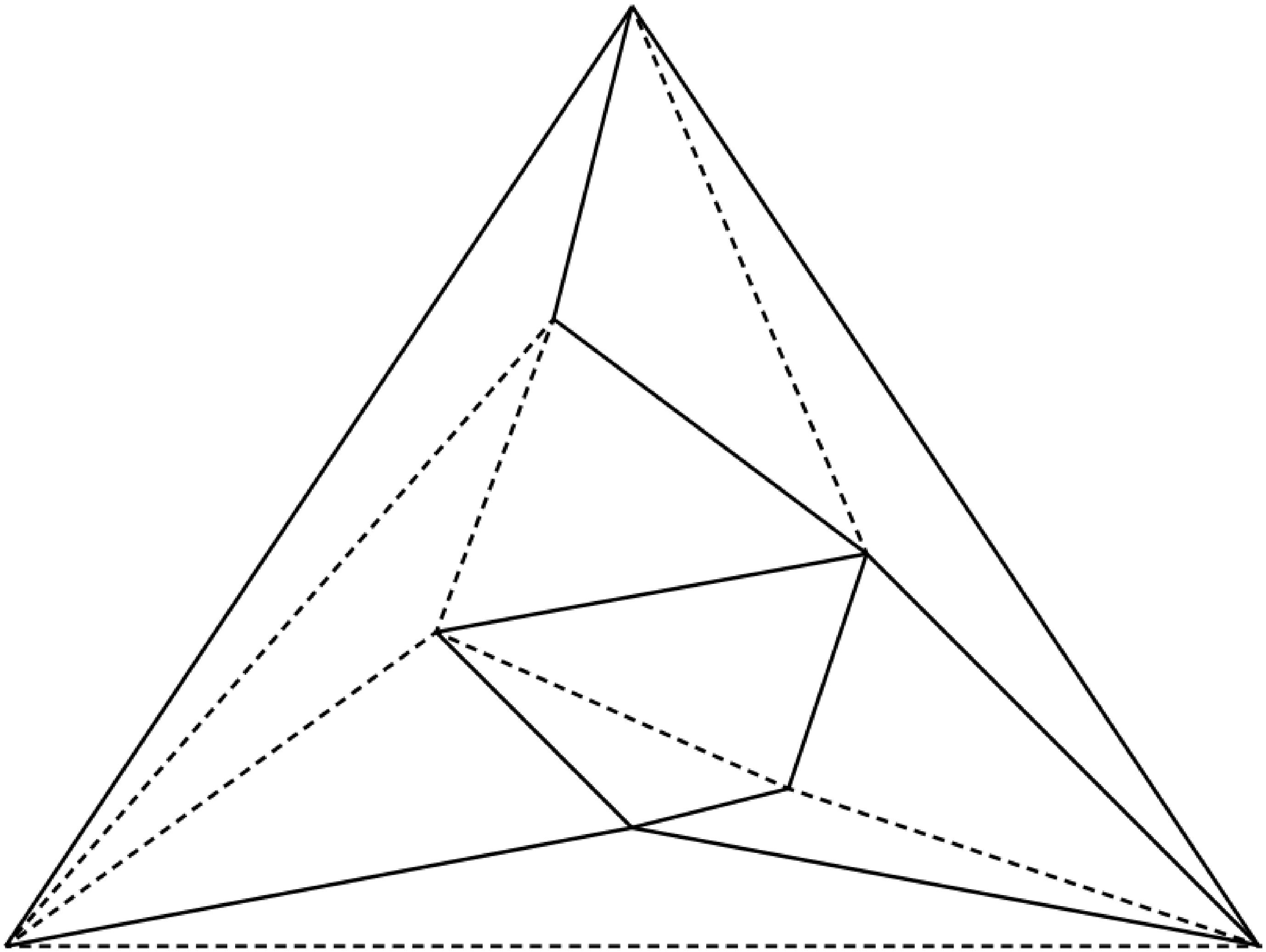


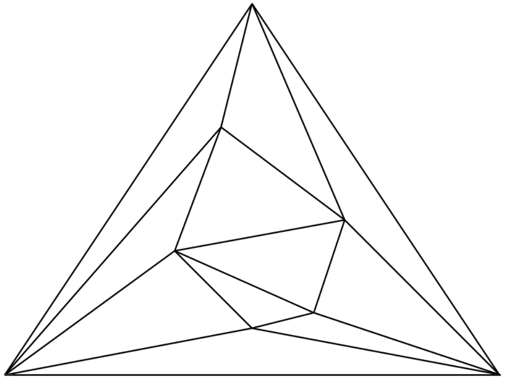


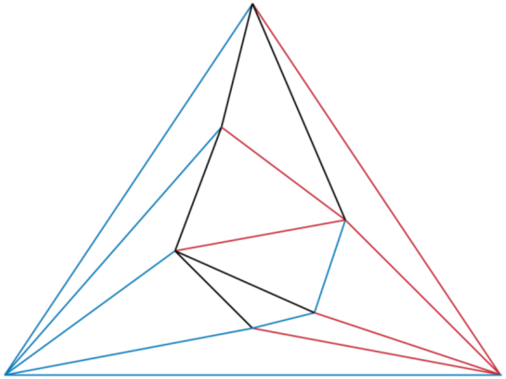
Planar Graphs



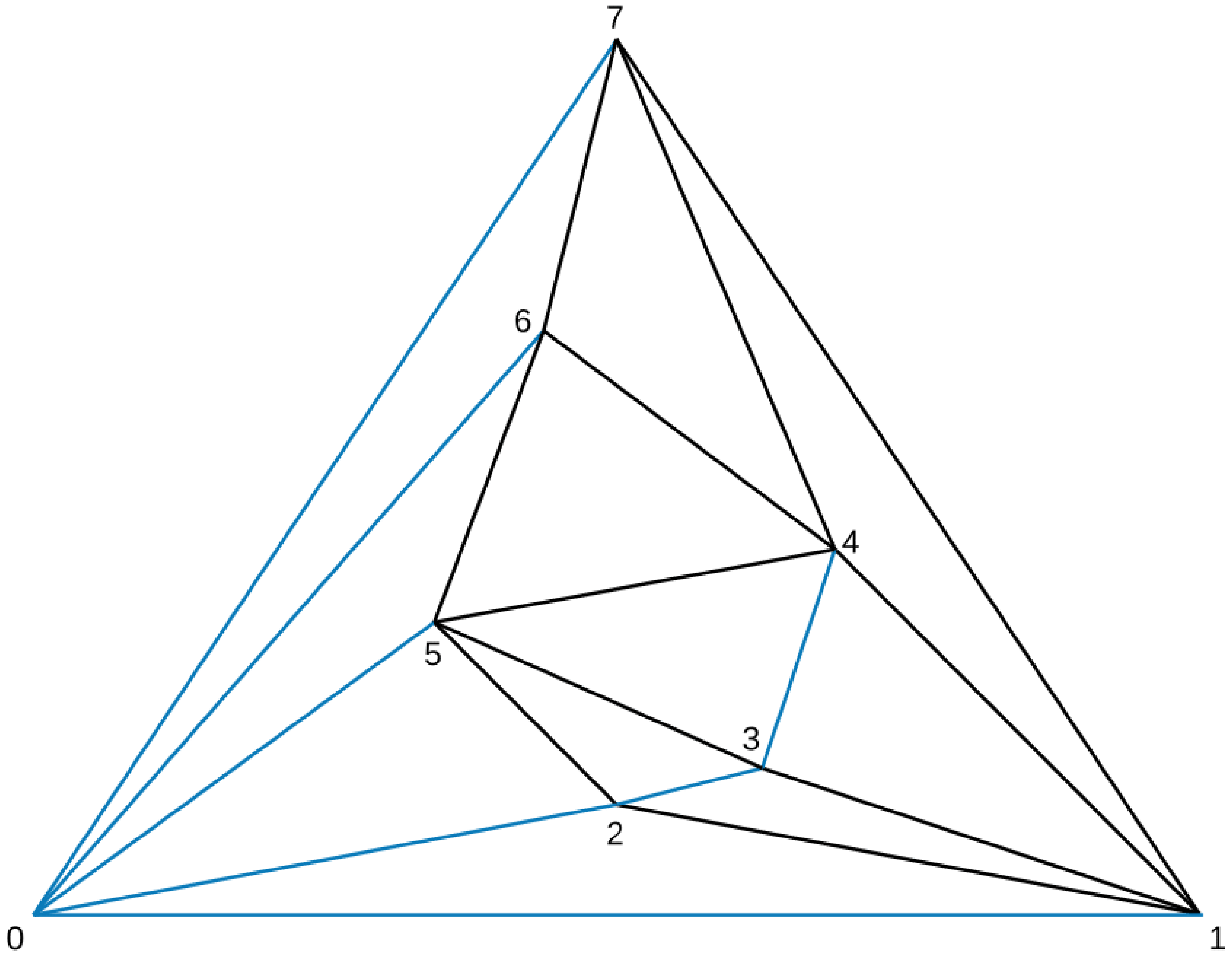
Triangulated Graphs



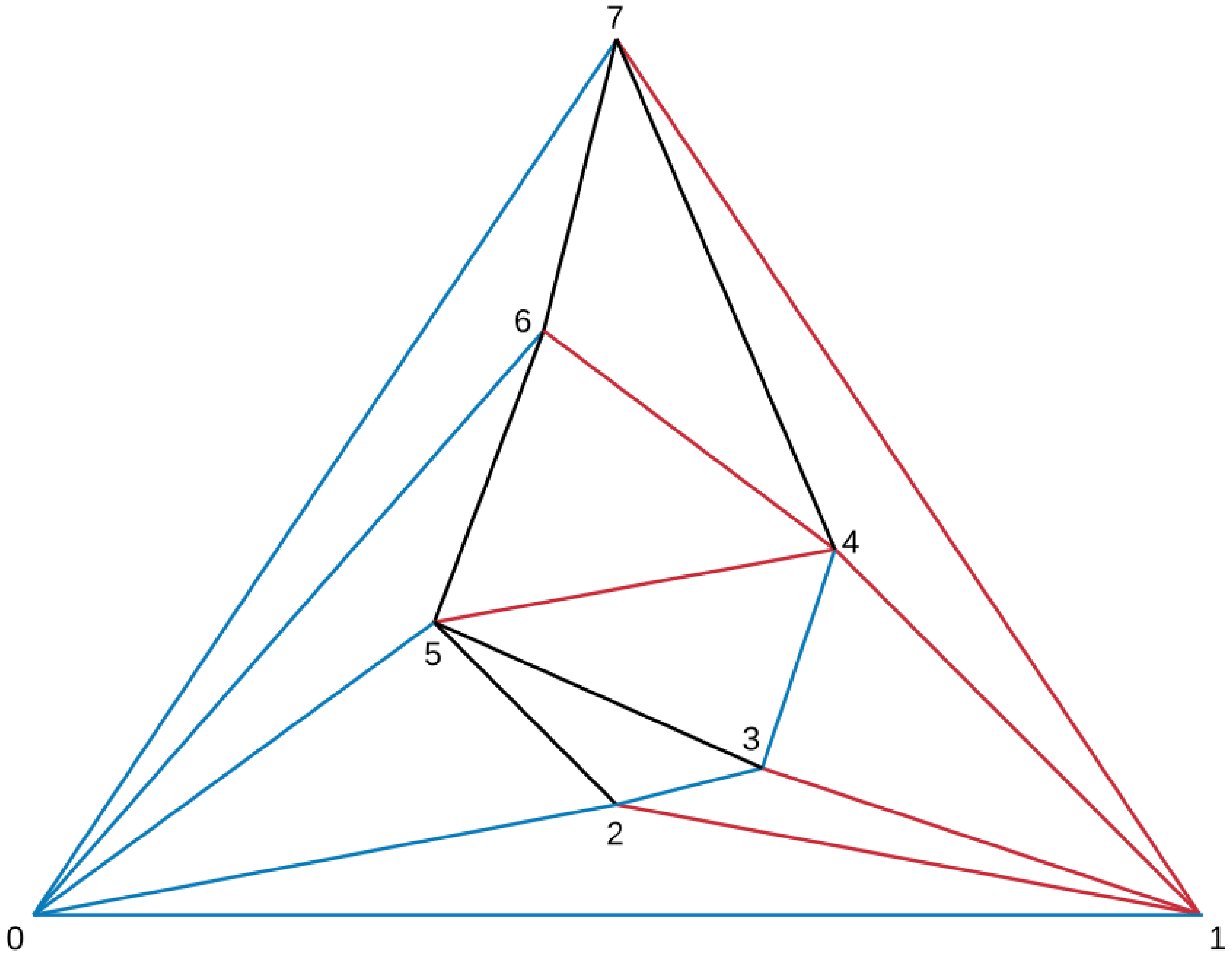




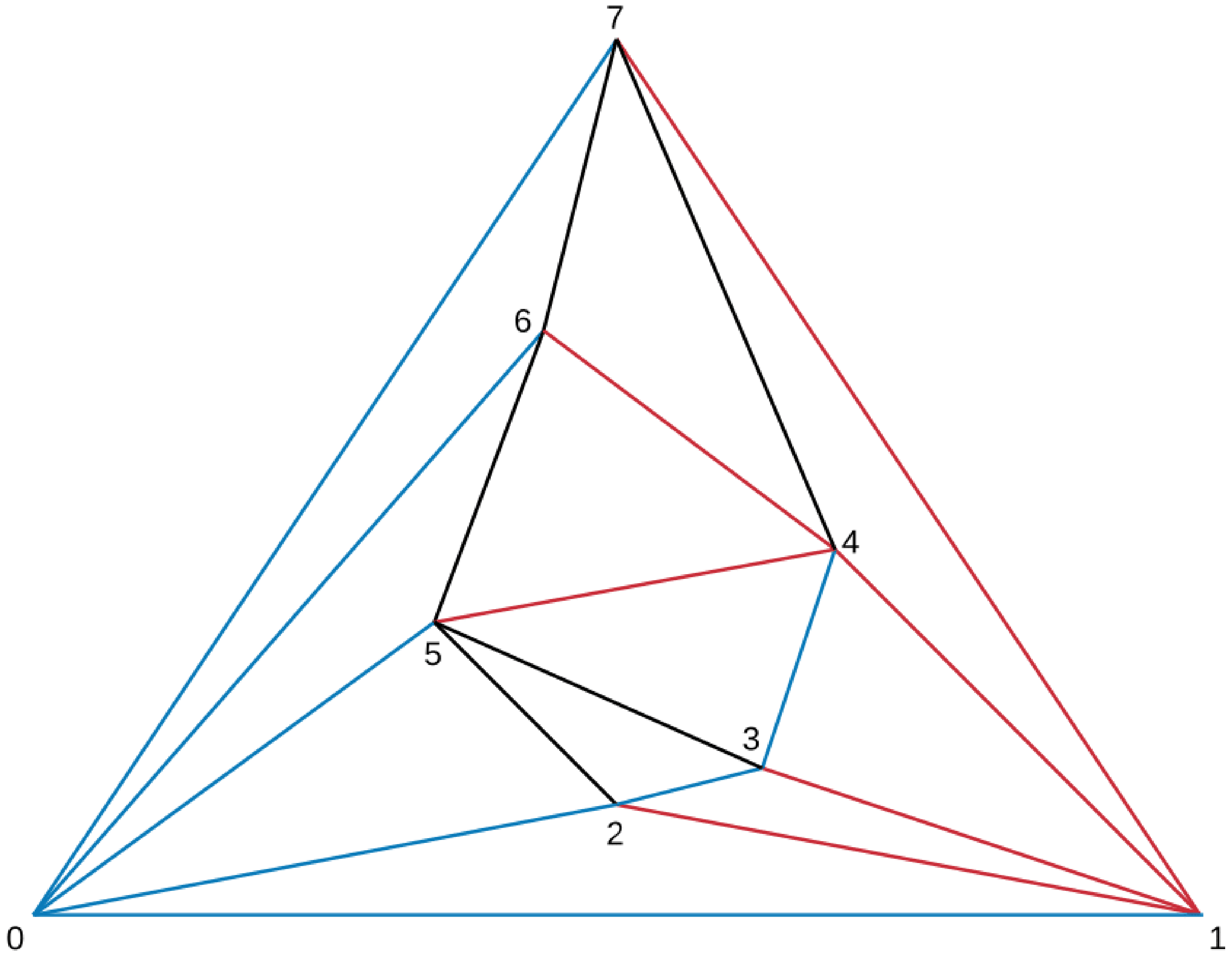
First Tree – Canonical Spanning Tree

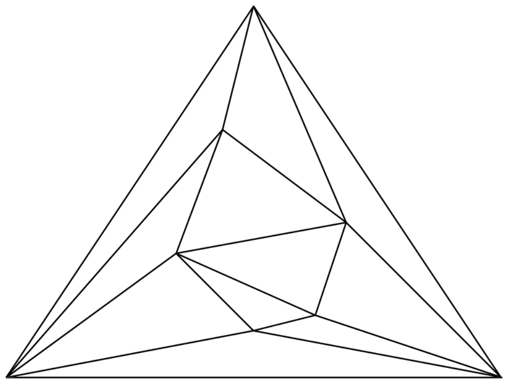


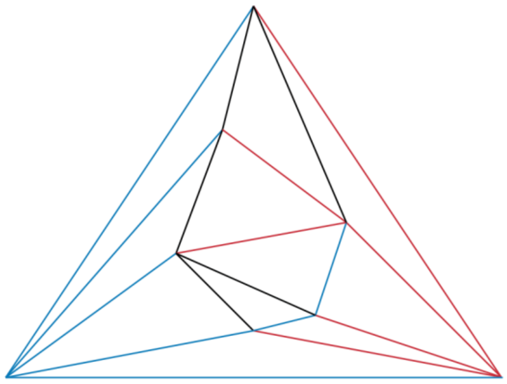
Second Tree



Third Tree





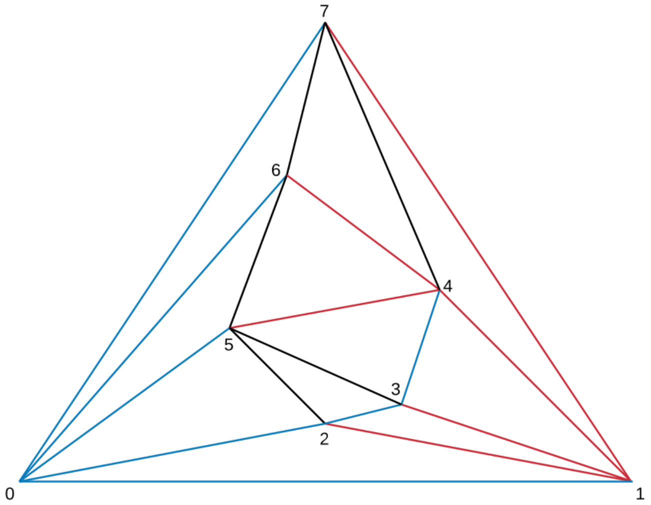


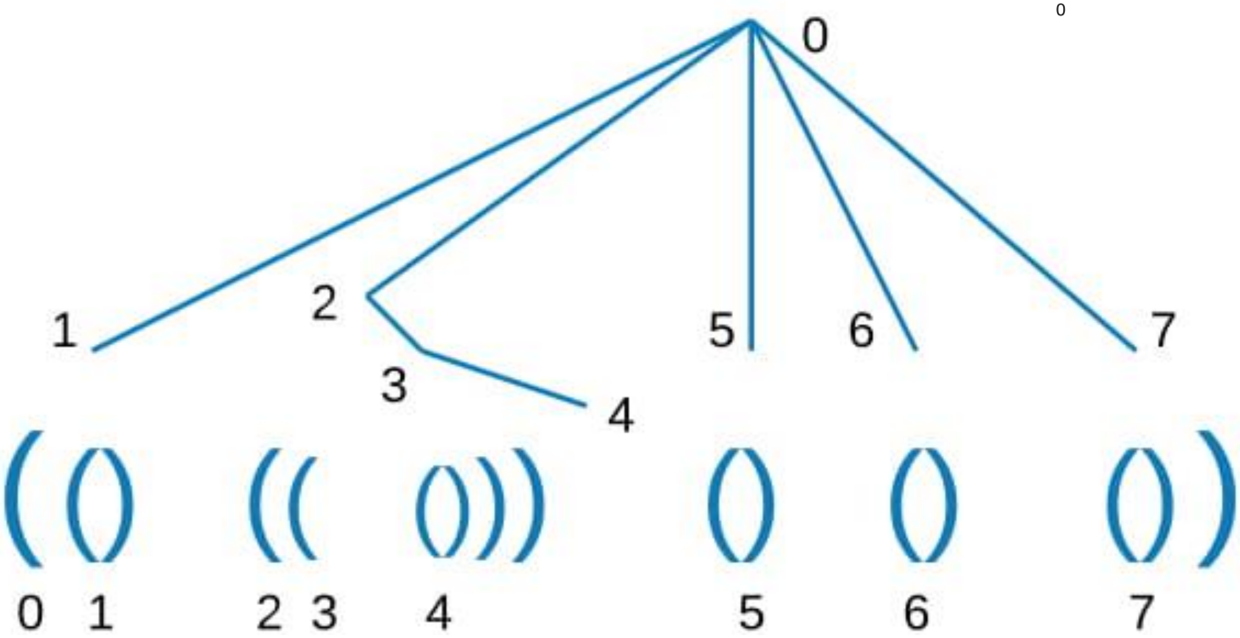
(() ((())) () () ())

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

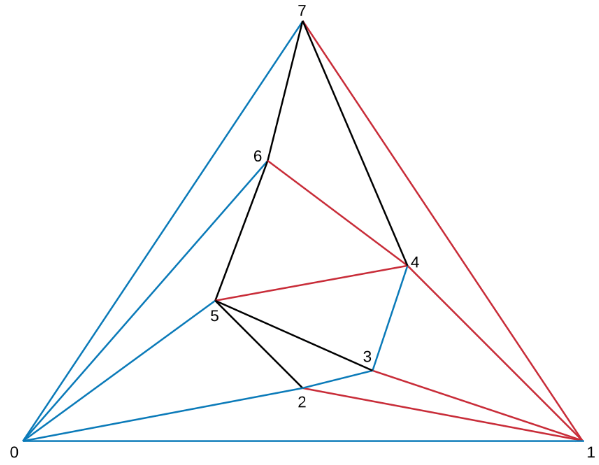
{ { { } } { } { } }

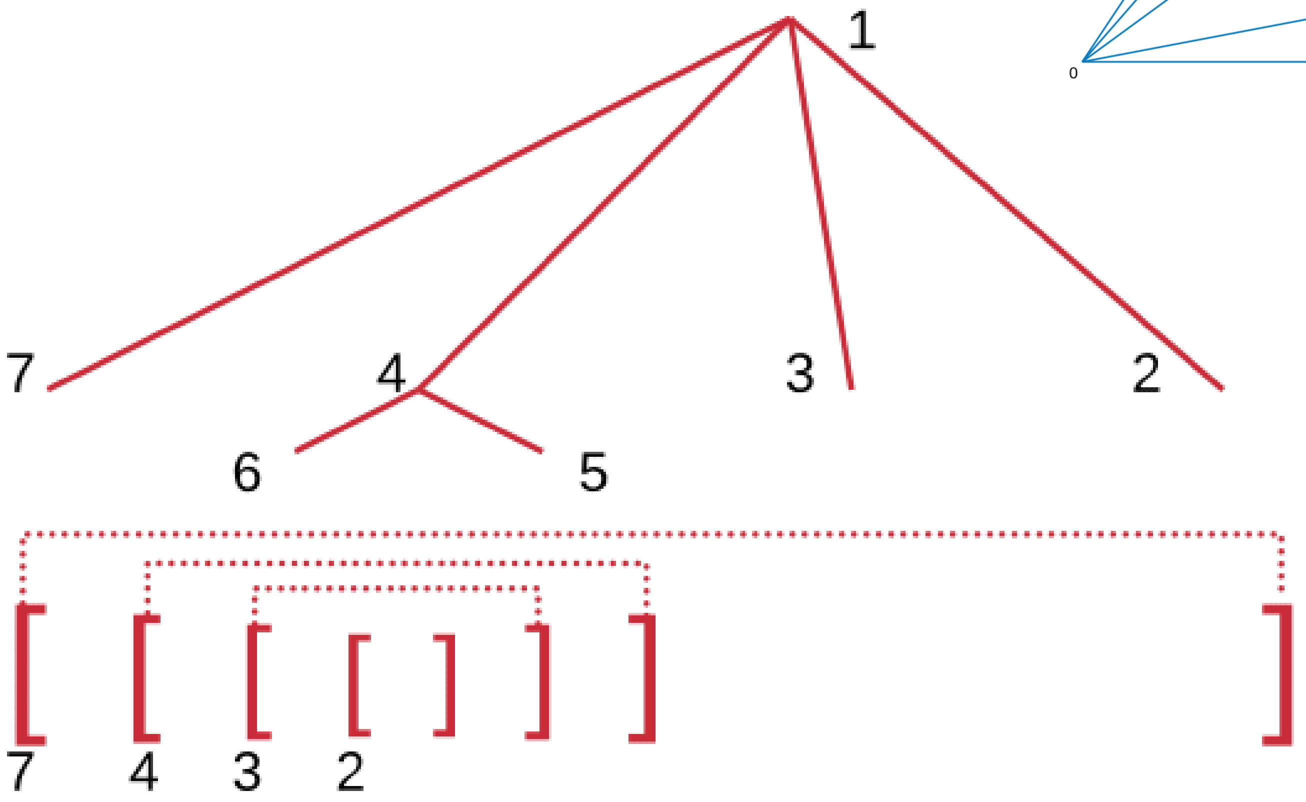
Parenthesis



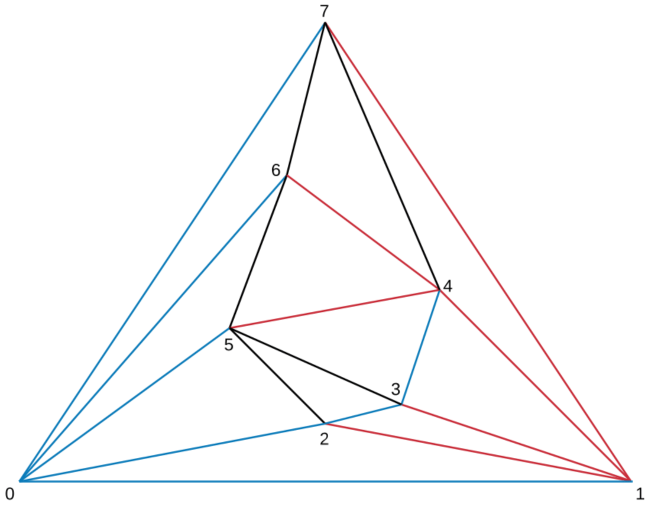


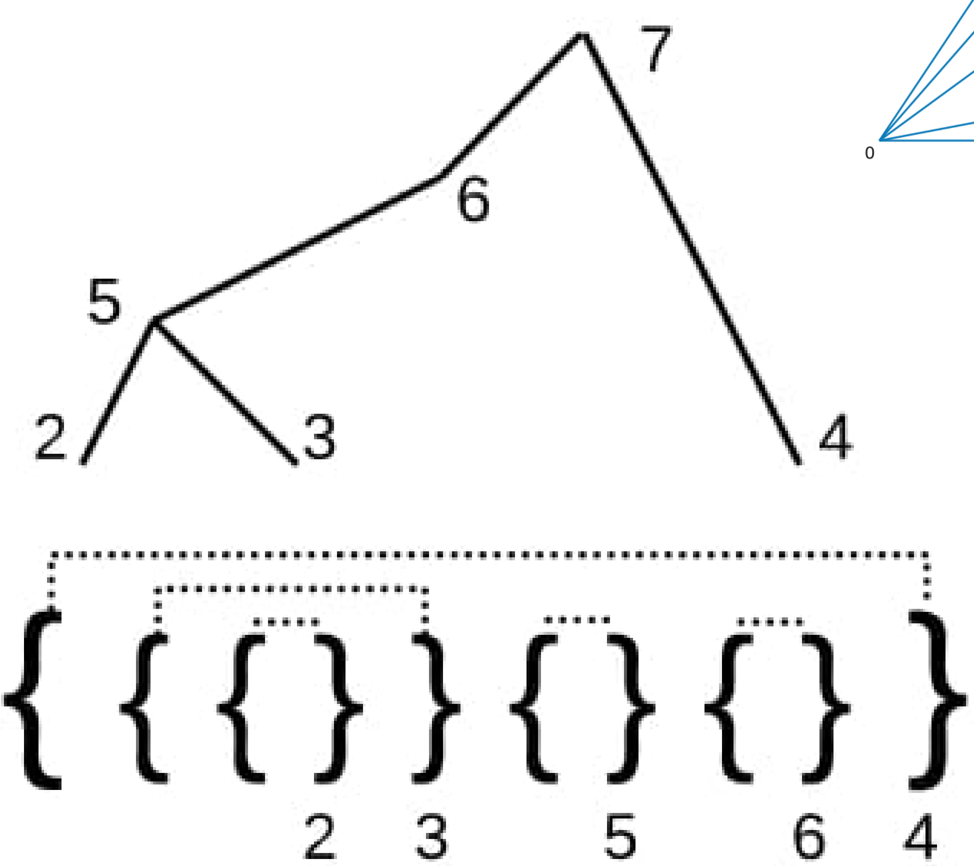
Parenthesis

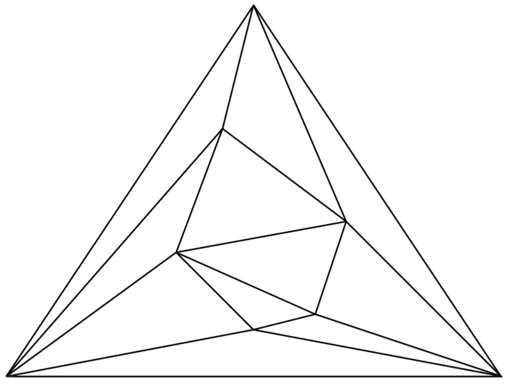


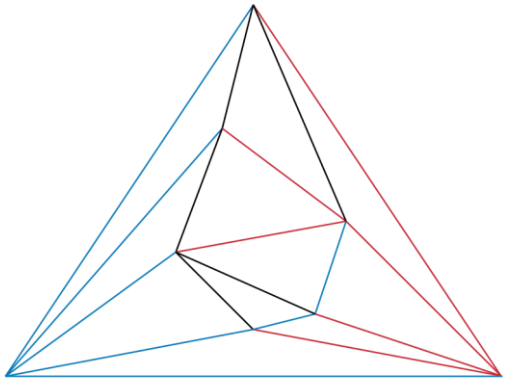


Parenthesis









(() ((())) () () ())

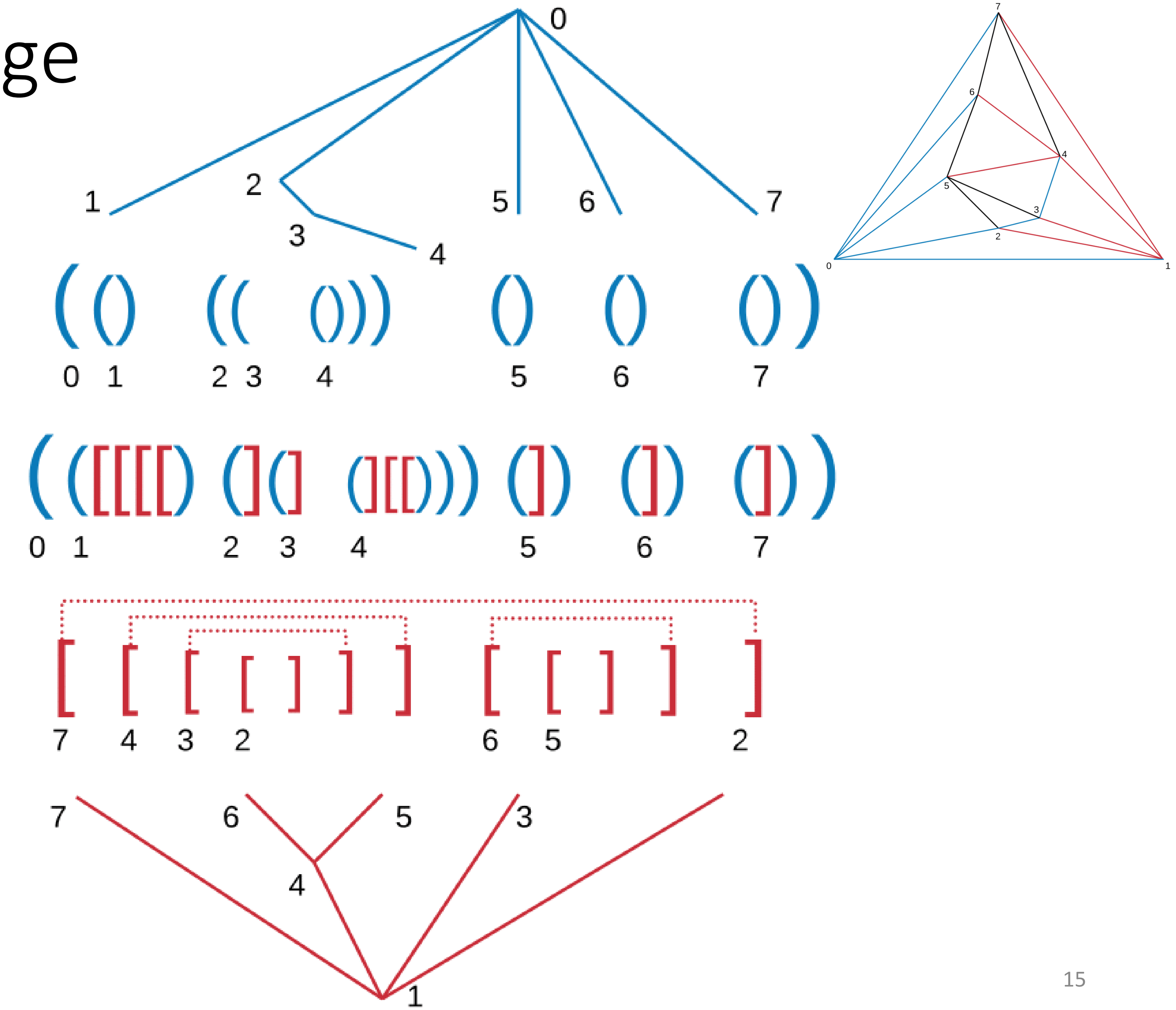
[[[[ ]]] [[ ]]]

{ { { } } { } { } }

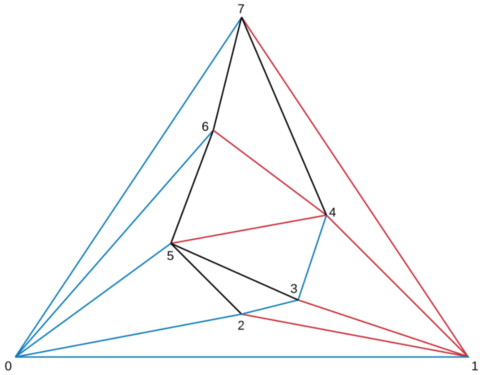
(([[[[) (](] (][[{){){) (}}]{) (}]{) (}}])))

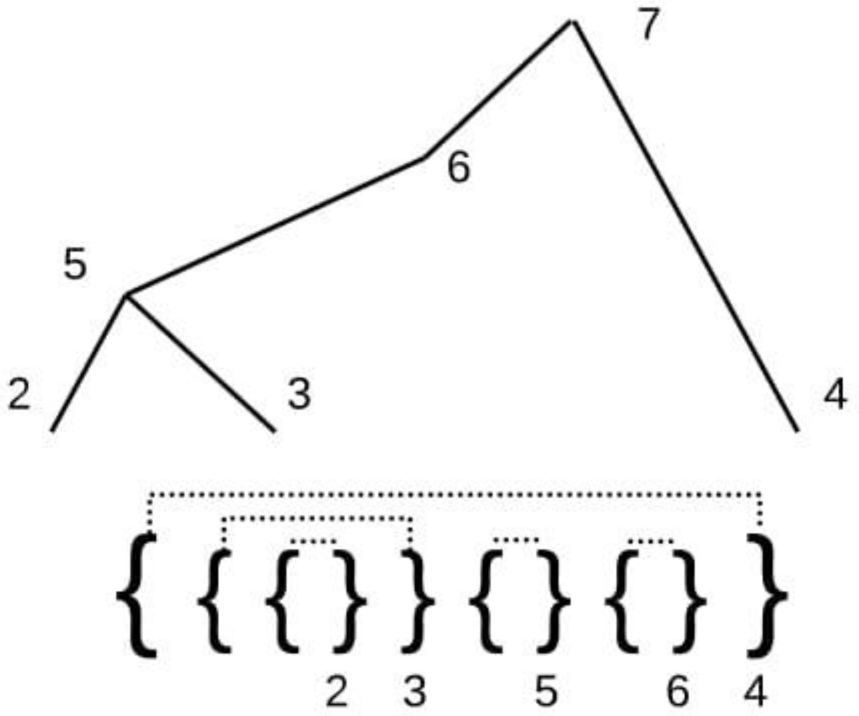
0 1 2 3 4 5 6 7

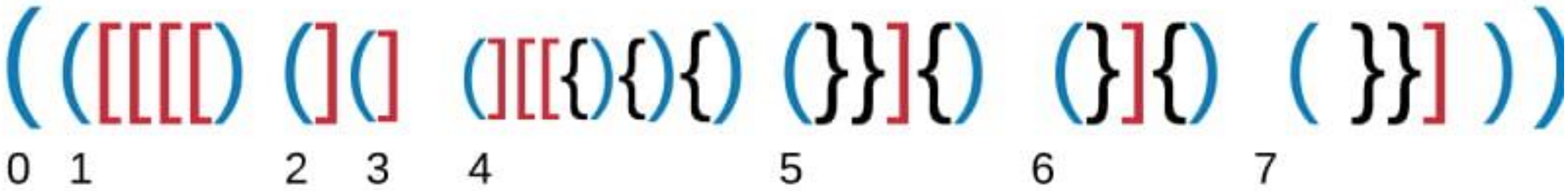
Merge



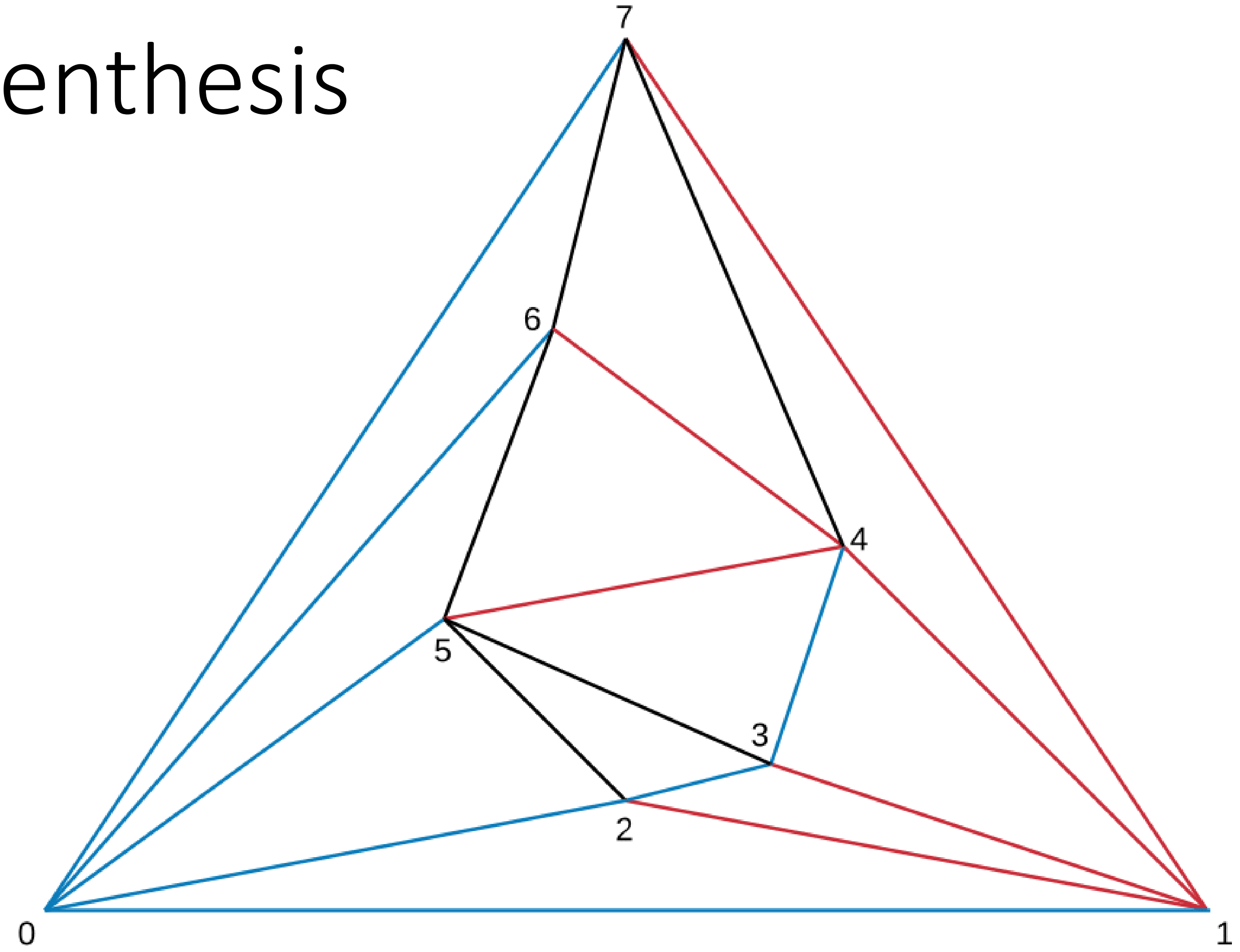
Merge







Parenthesis



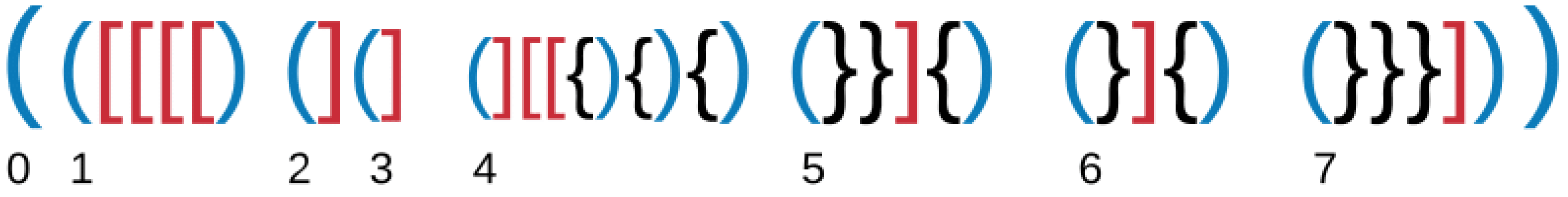
(([[[[) (]( ] (][[{){){) (}}]{) (}]{) (}}}])))

0 1 2 3 4 5 6 7

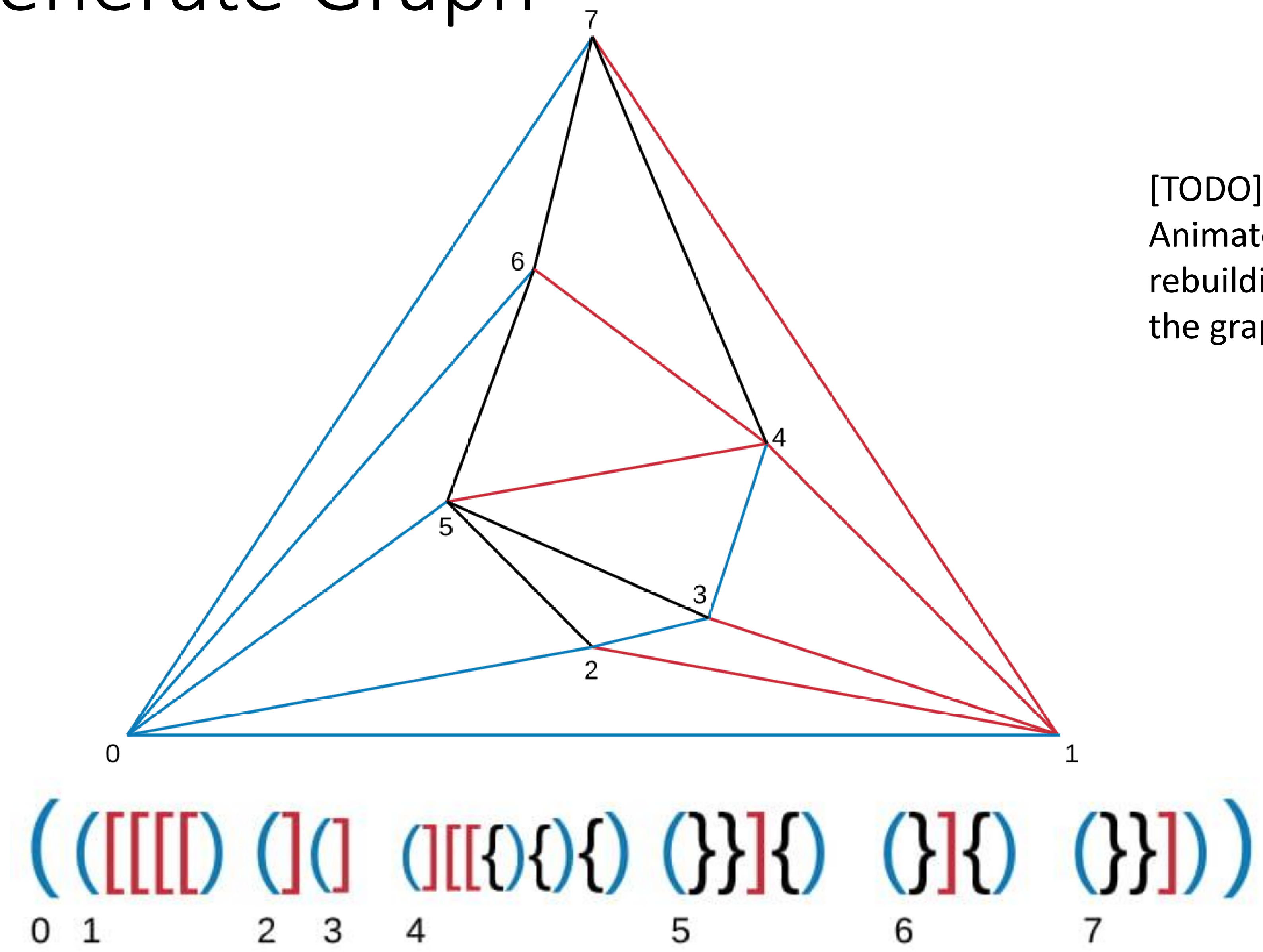
Parenthesized representation

- Each type -> forms a correct parenthesis
- Two parenthesis per Edge

=> 2m*3 Bits necessary

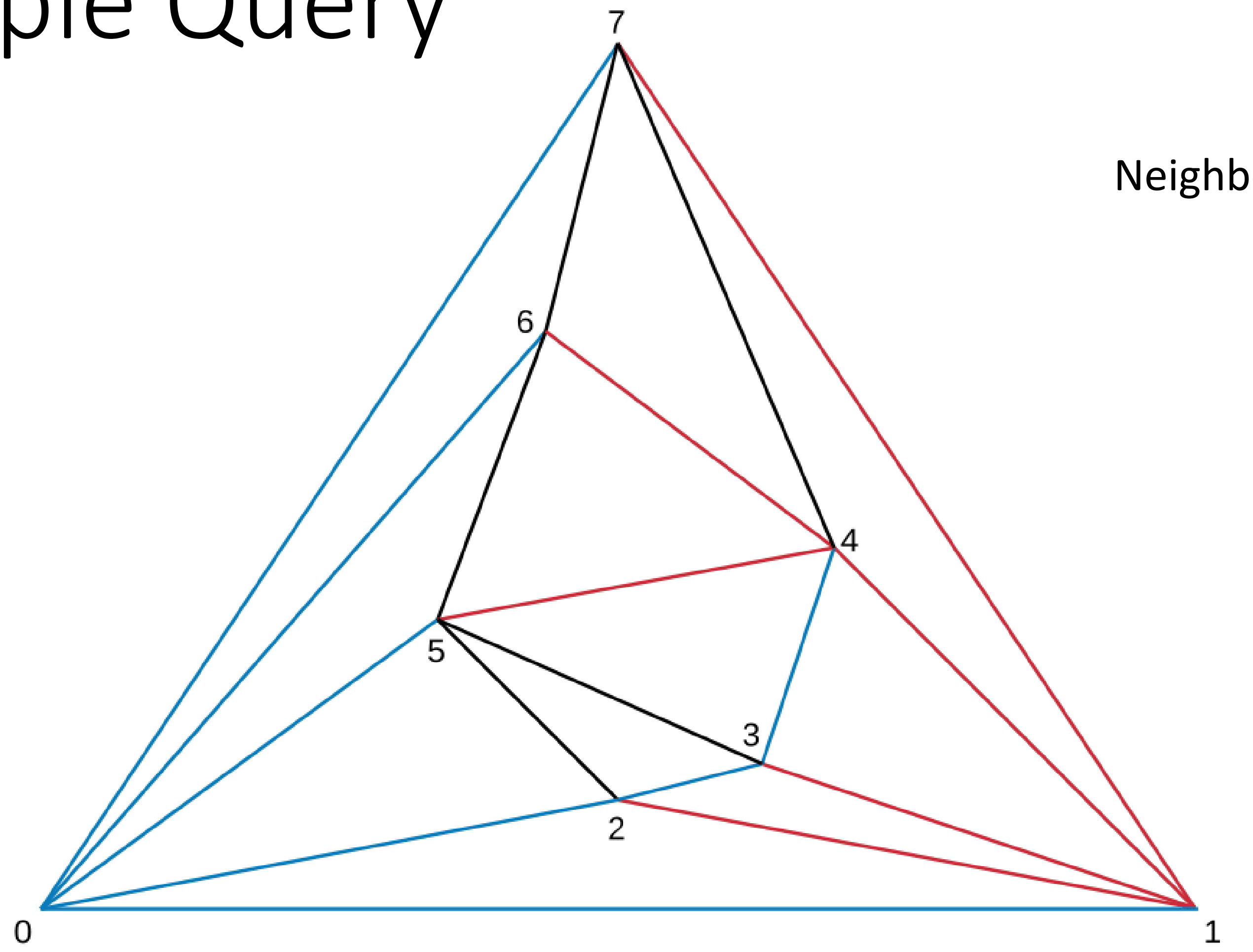


Generate Graph

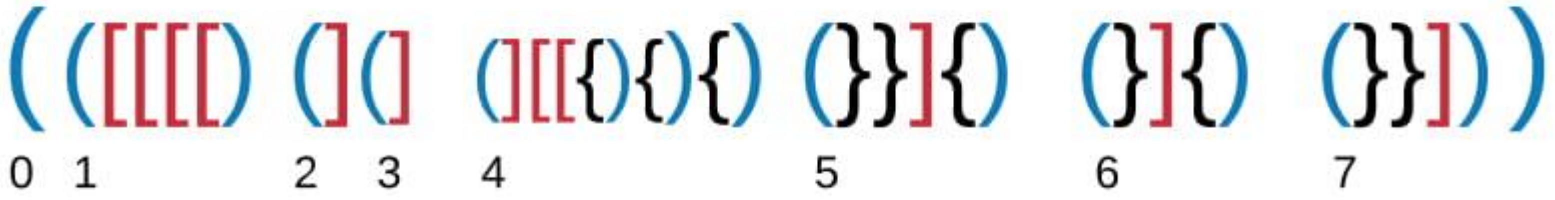


[TODO]
Animated
rebuilding
the graph

Sample Query



Neighborhood Query



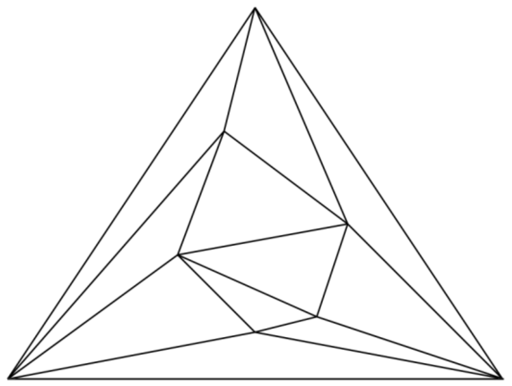
Parenthesized representation

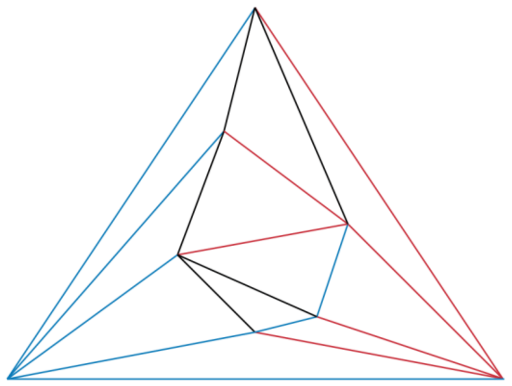
- Why to use the parentheses
- Explain Results and supported functions

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 1      2 3   4          5      6     7
```

⇒ 6m Bits necessary

⇒ Query runtimes like uncompressed

End

Thank you for your attention!