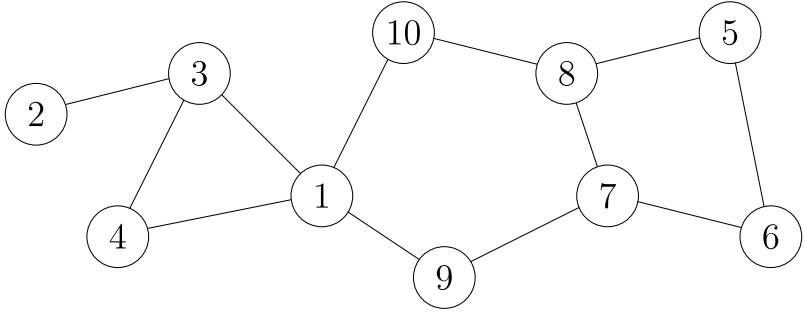
Karlsruhe Institute of Technology

CCH Visualization

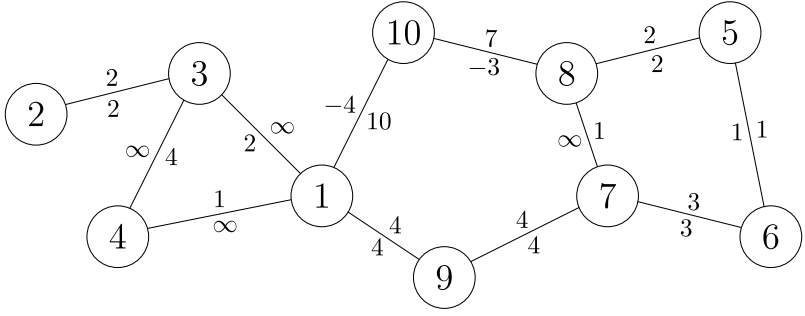
Patrick Steil, Daniel-Delong Zhang | March 13, 2024

Original Graph



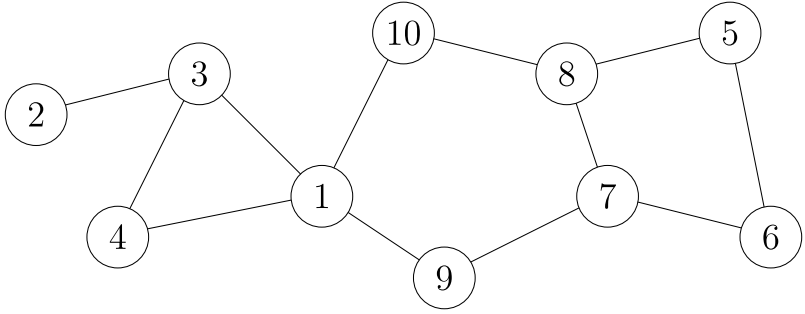
The original graph this visualization will showcase.

Original Graph

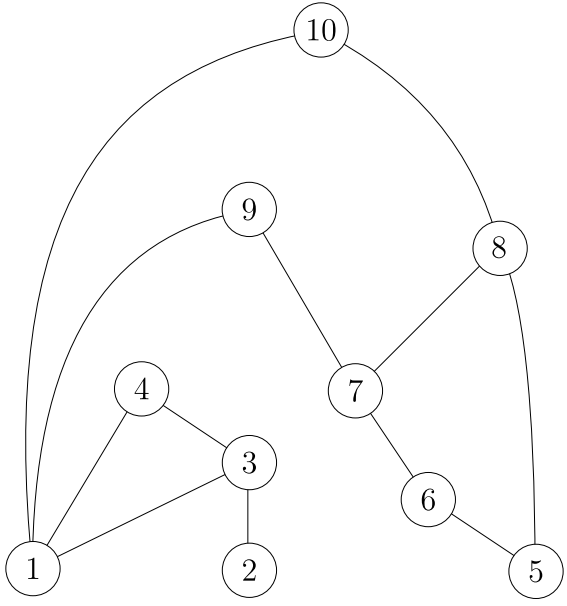


The respective weight is on the right side of the edge.

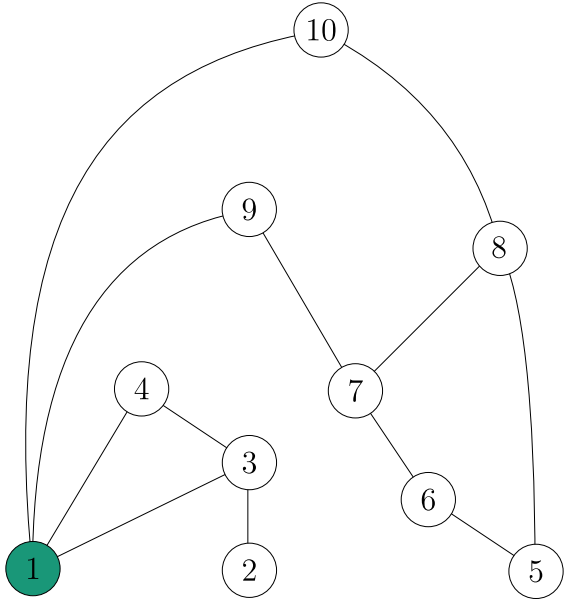
Contraction



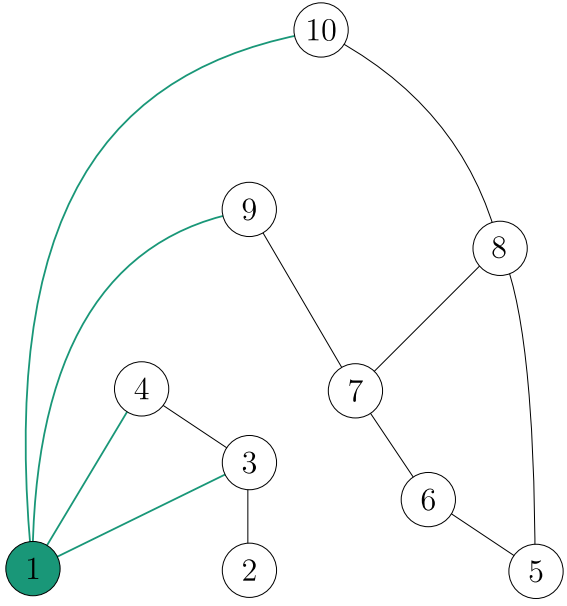
Contraction on the undirected graph without metrics.



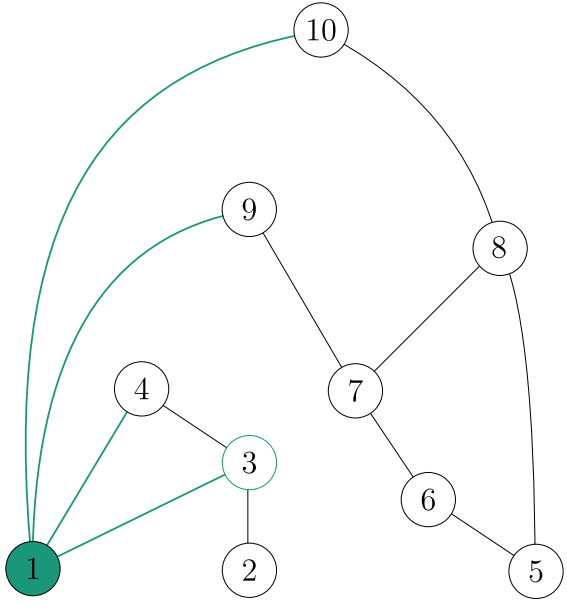
Ordered Graph



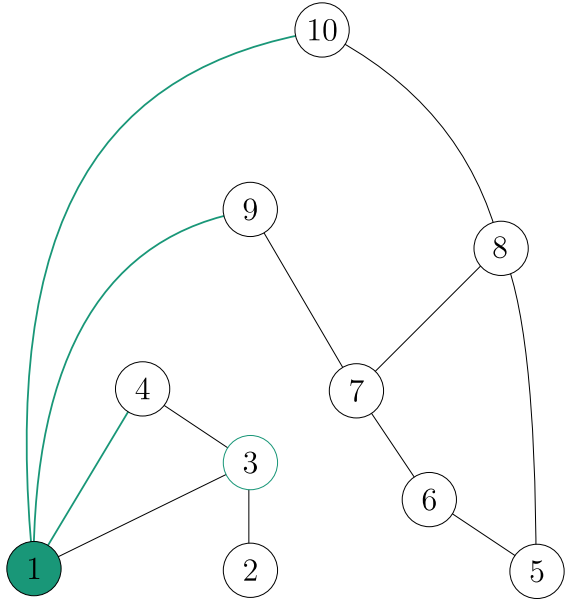
Contract vertex 1



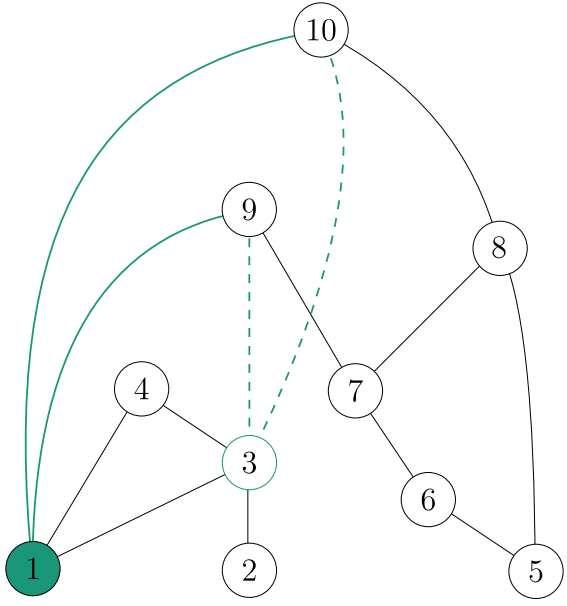
Find the outgoing edges

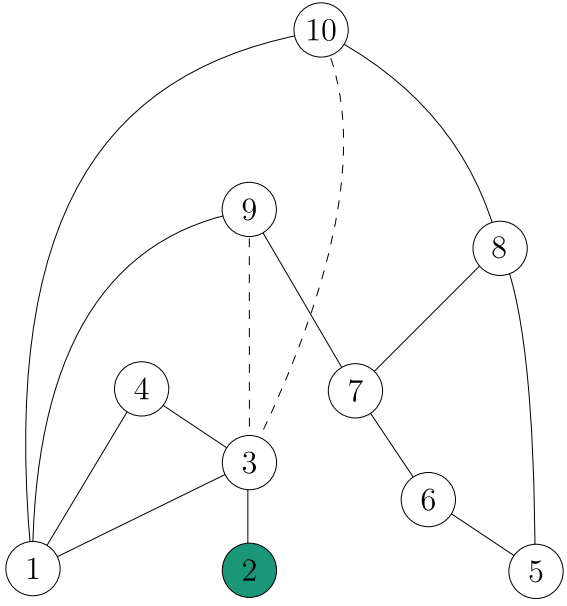


Set the smallest one as the parent

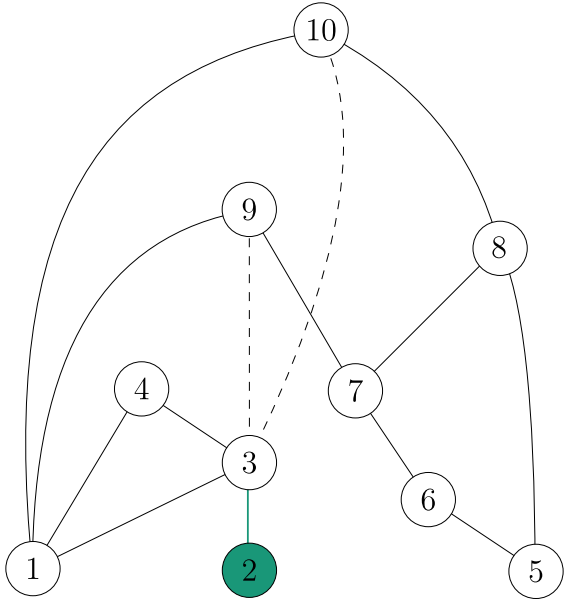


Copy the remaining edges to the parent if it is possible

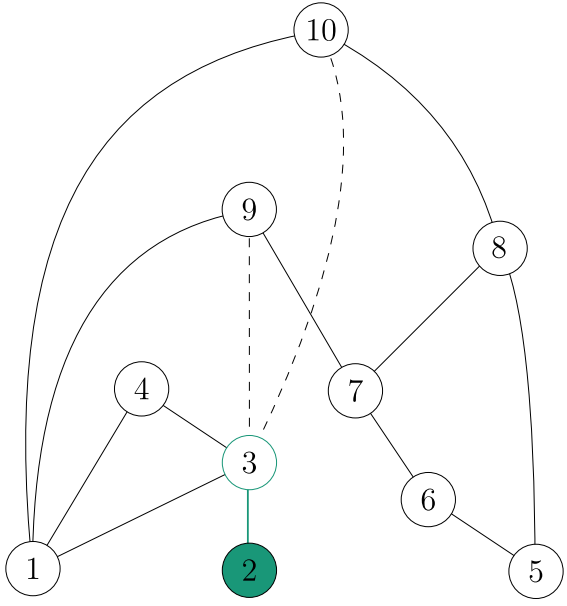

10
9
8
4
7
3
6
1
2
5



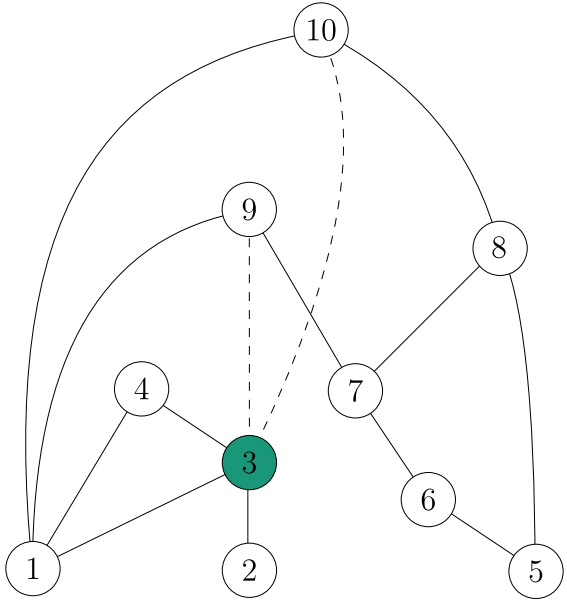
Contract vertex 2



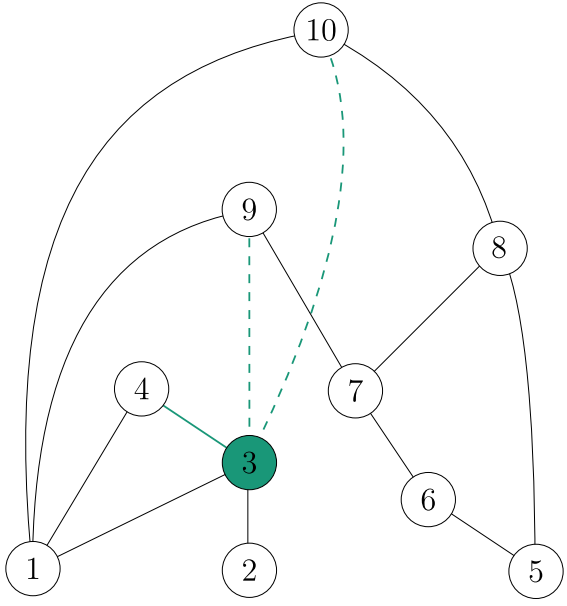
Find the outgoing edges



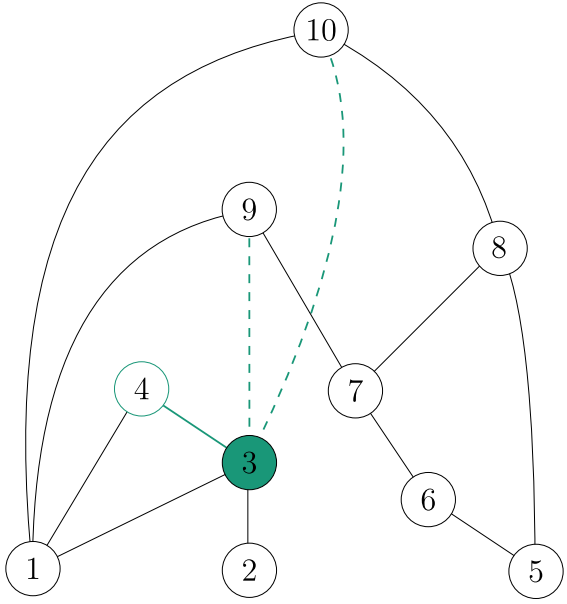
Set the smallest one as the parent



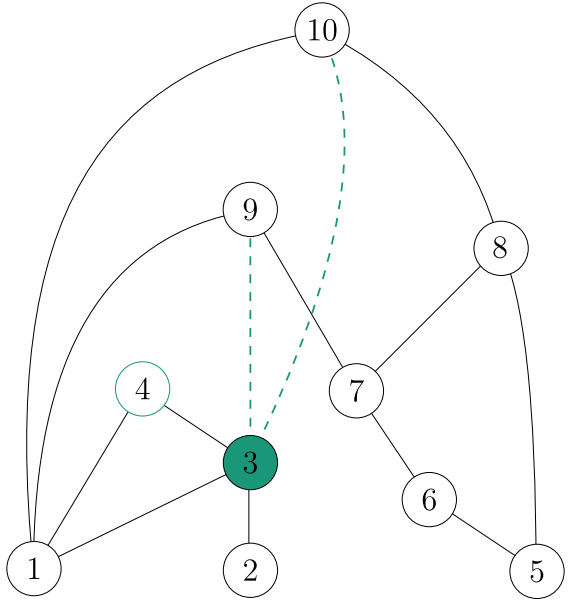
Contract vertex 3



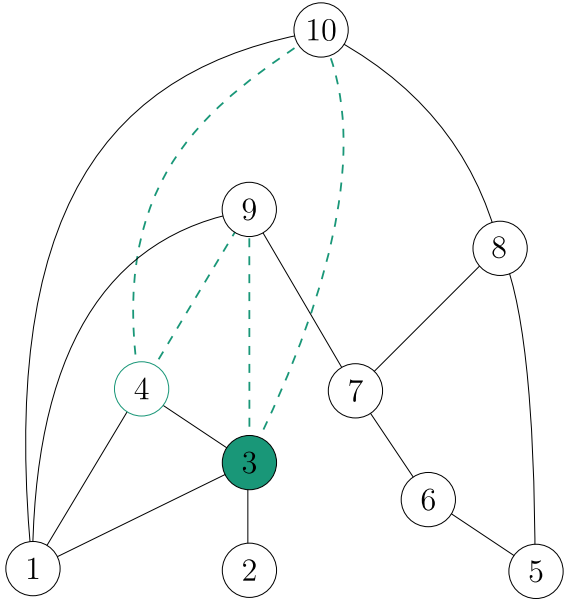
Find the outgoing edges

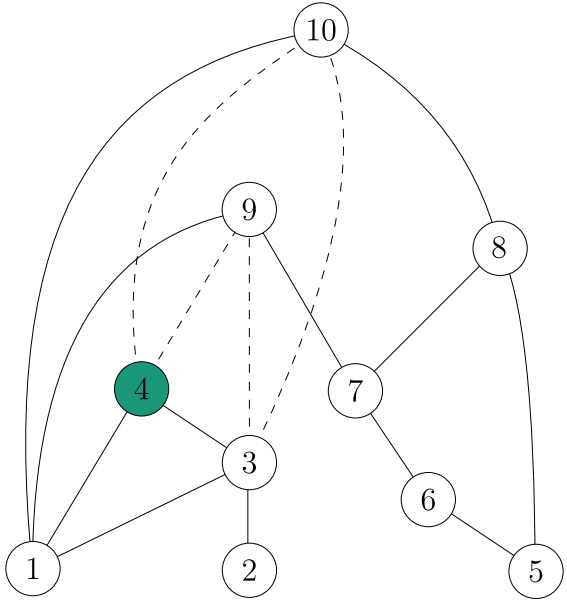


Set the smallest one as the parent

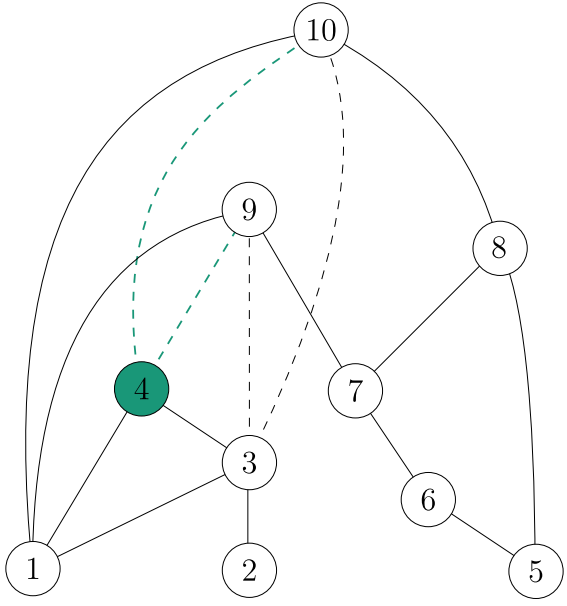


Copy the remaining edges to the parent if it is possible

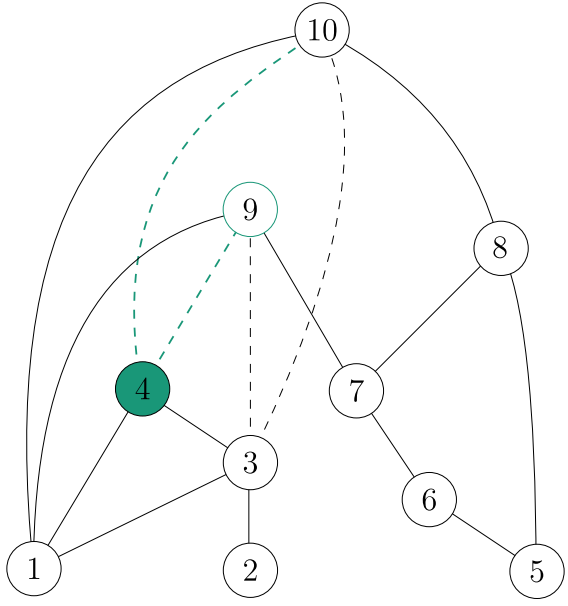

10
9
8
4
7
3
6
1
2
5



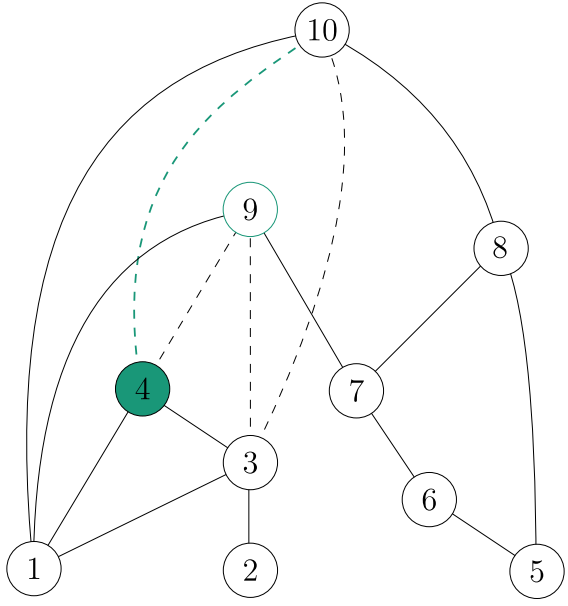
Contract vertex 4



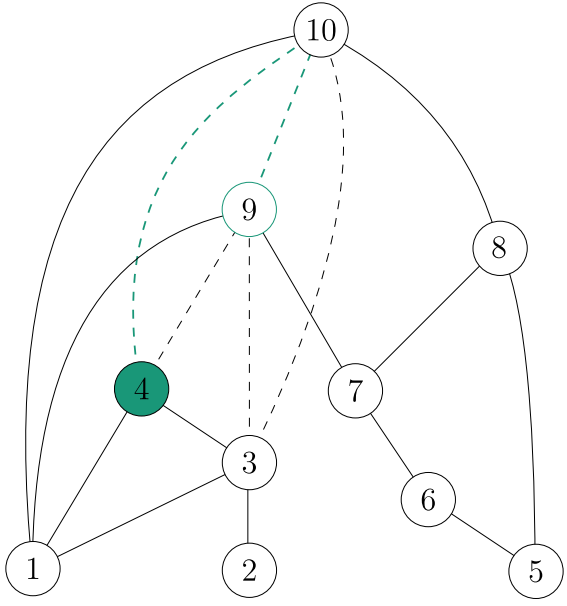
Find the outgoing edges

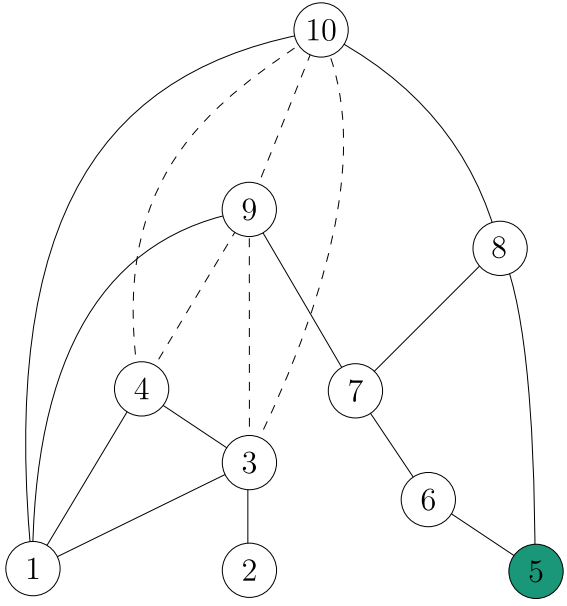


Set the smallest one as the parent

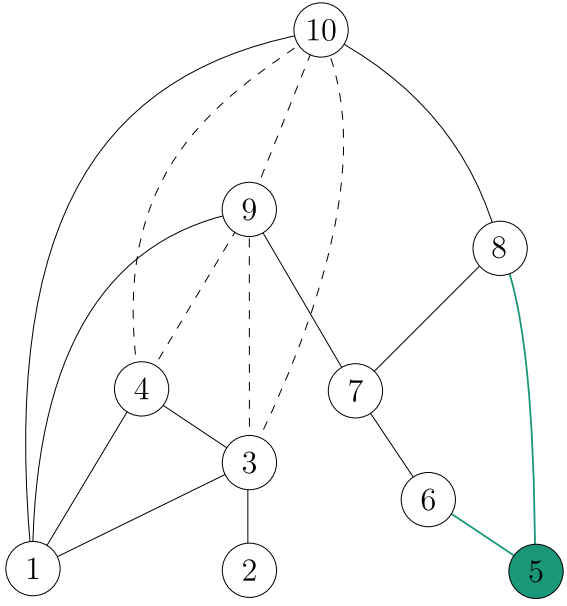


Copy the remaining edges to the parent if it is possible

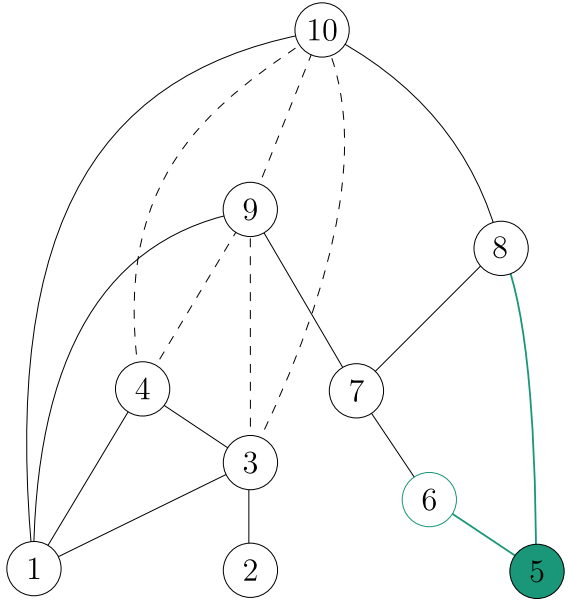

10
9
8
4
7
3
6
1
2
5



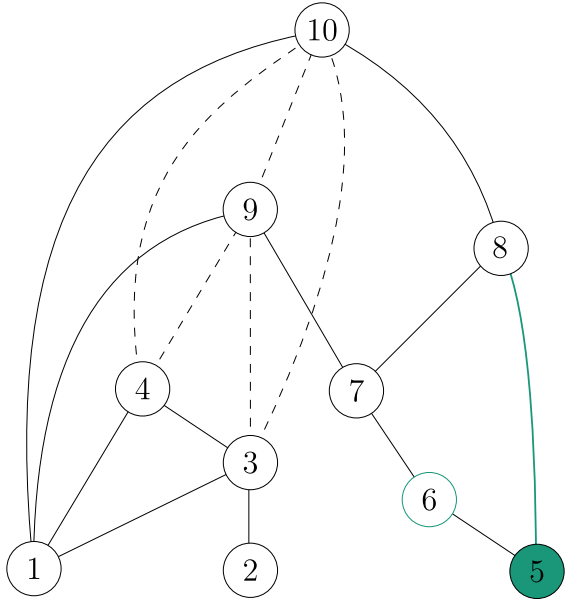
Contract vertex 5



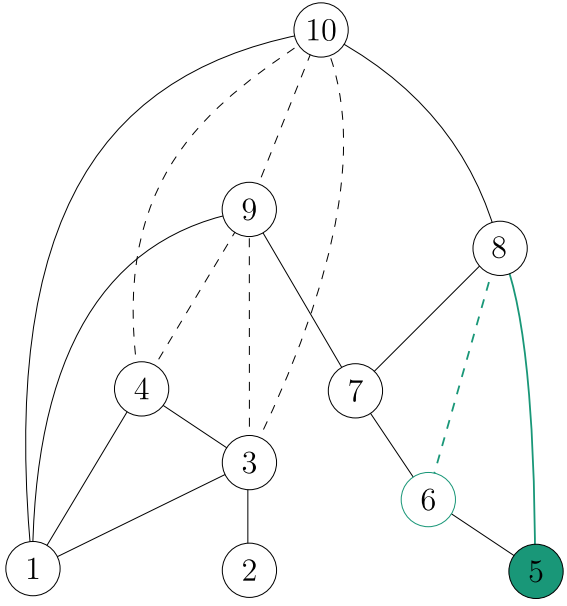
Find the outgoing edges

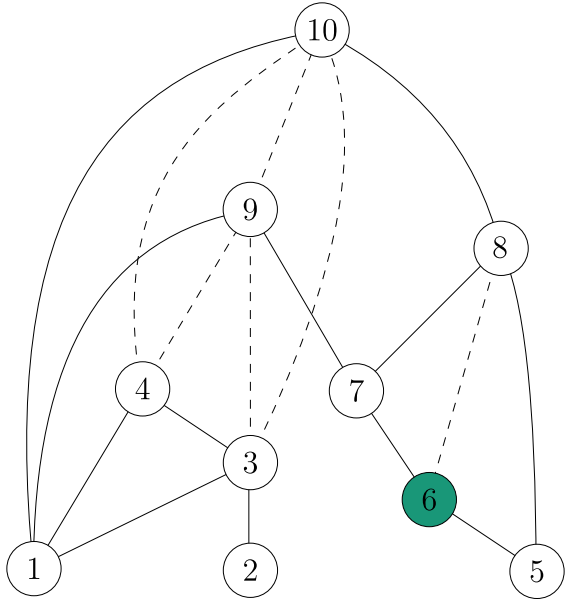


Set the smallest one as the parent

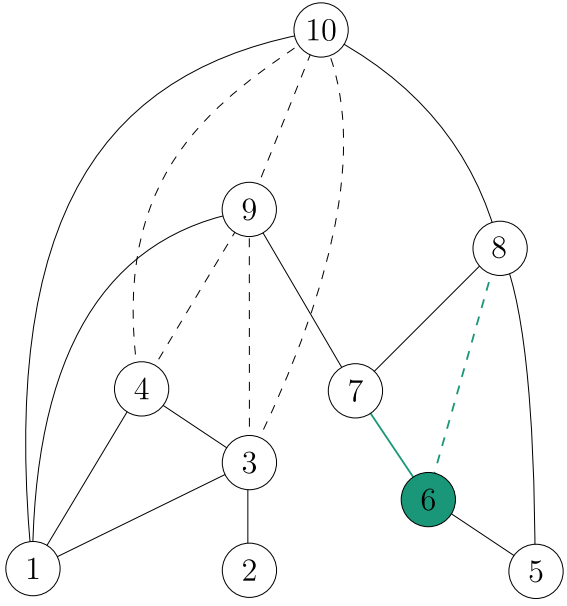


Copy the remaining edges to the parent if it is possible

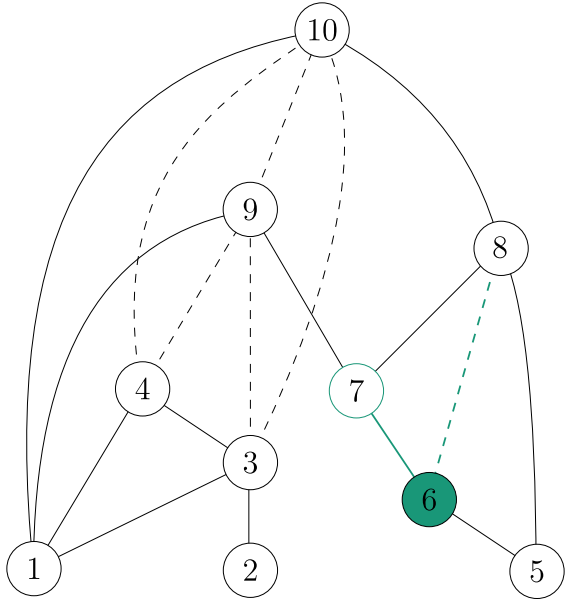

10
9
8
4
7
3
6
1
2
5



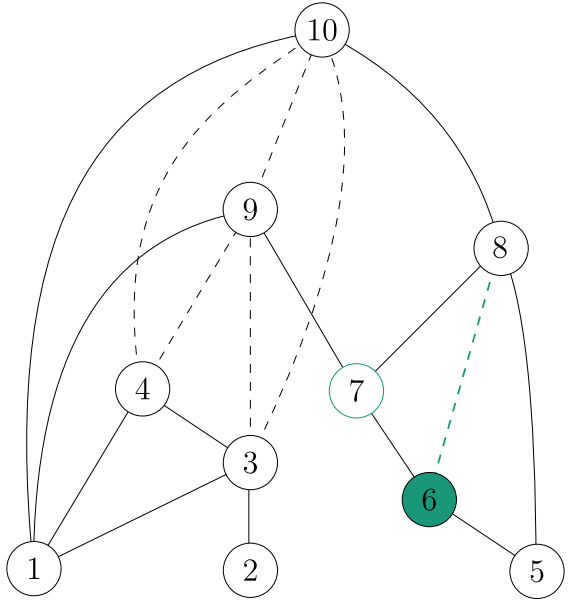
Contract vertex 6



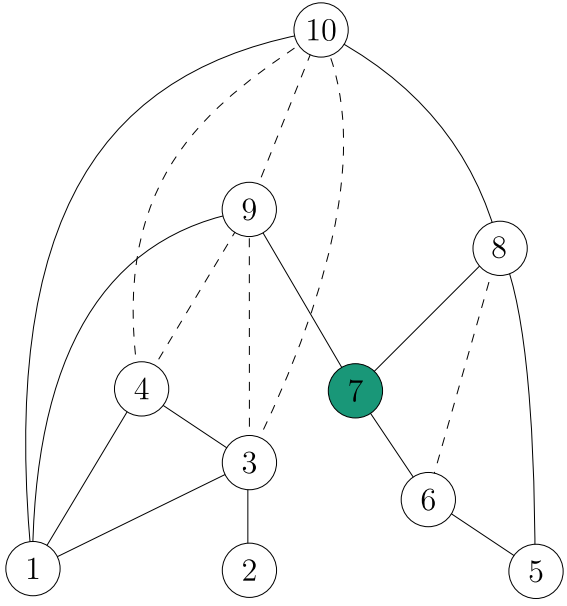
Find the outgoing edges



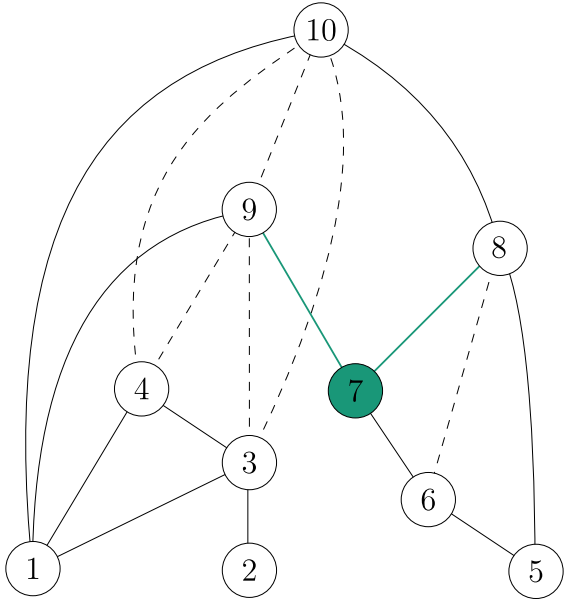
Set the smallest one as the parent



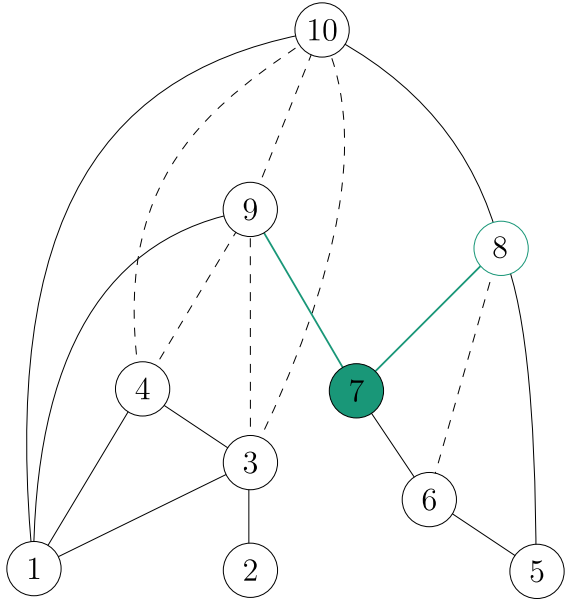
Copy the remaining edges to the parent if it is possible



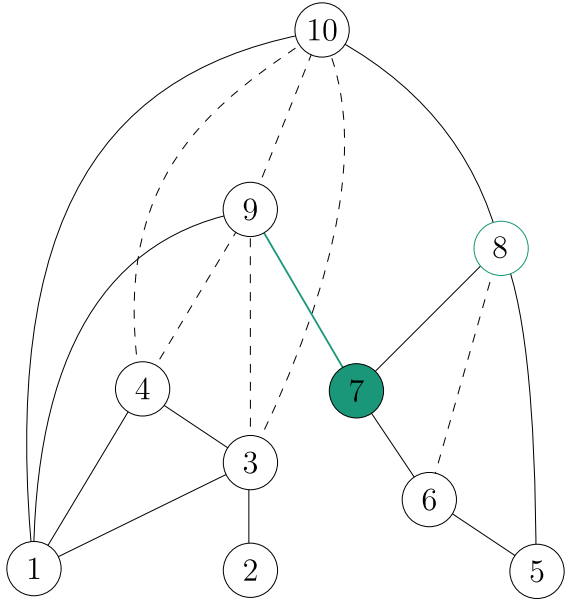
Contract vertex 7



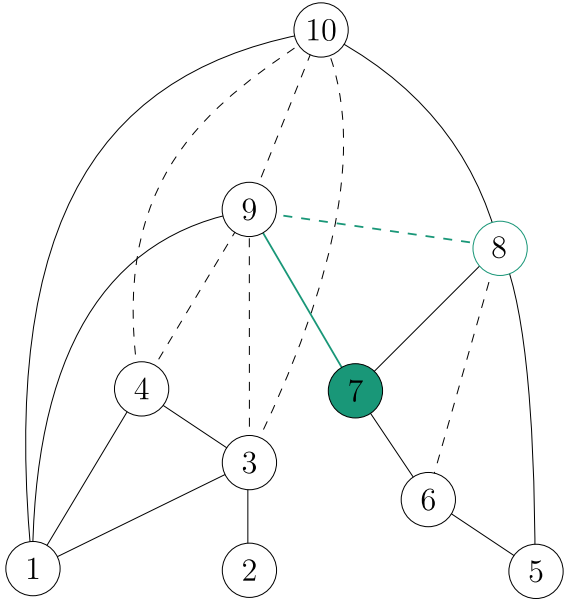
Find the outgoing edges

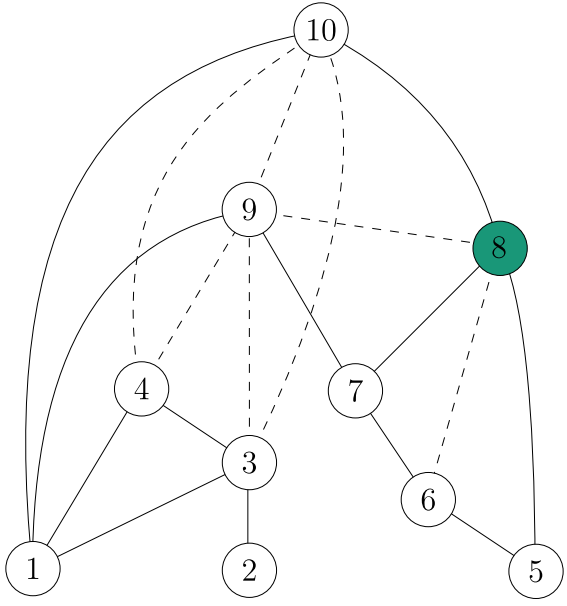


Set the smallest one as the parent

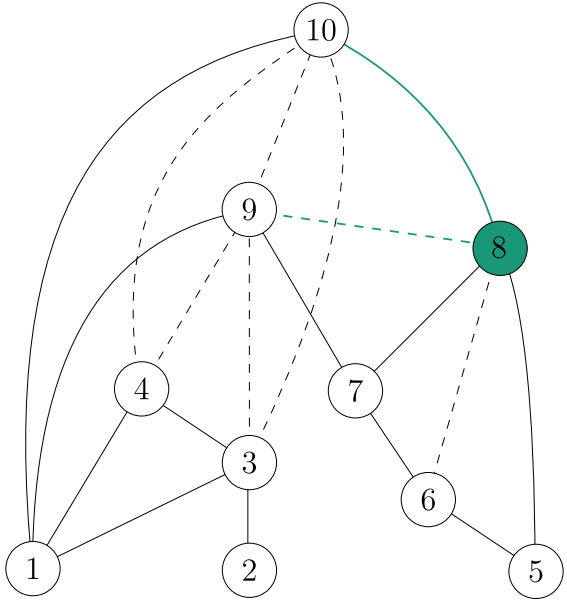


Copy the remaining edges to the parent if it is possible

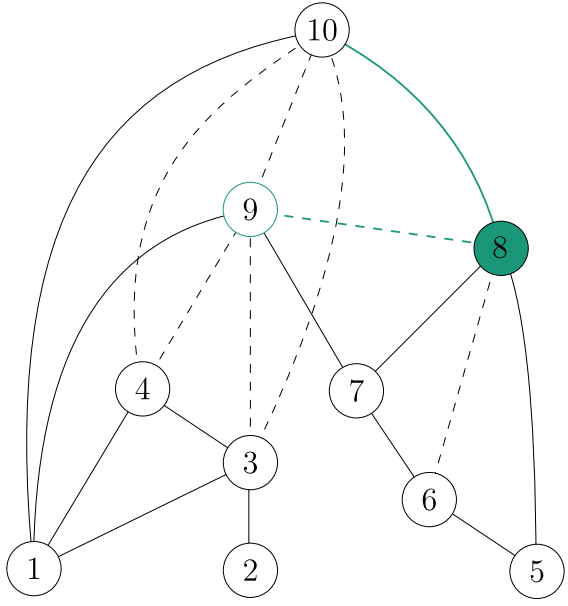

10
9
8
4
7
3
6
1
2
5



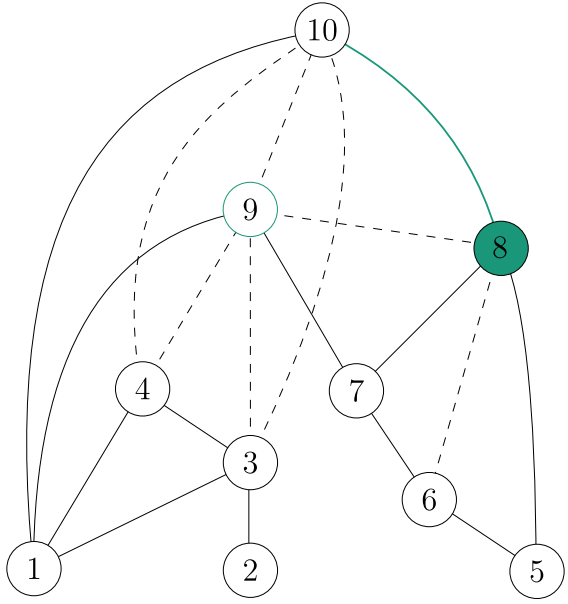
Contract vertex 8



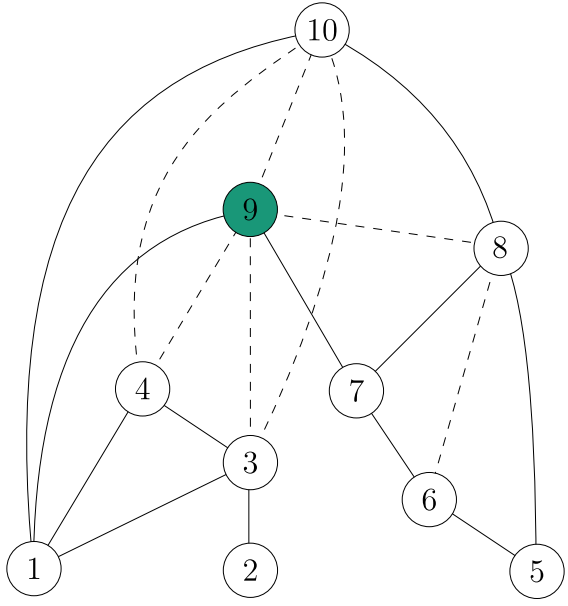
Find the outgoing edges



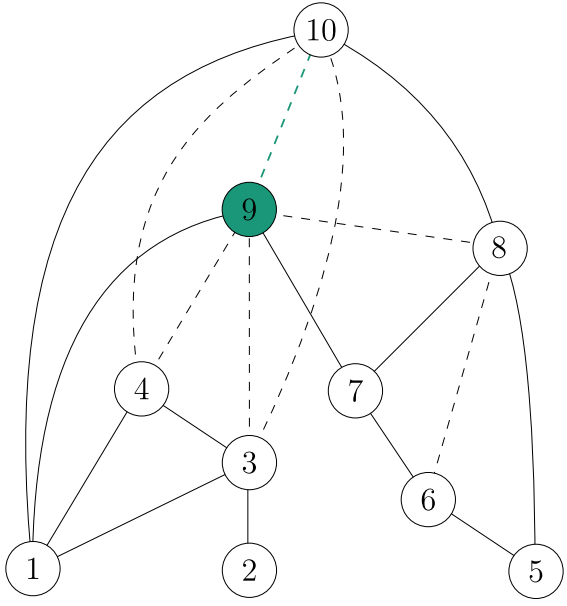
Set the smallest one as the parent



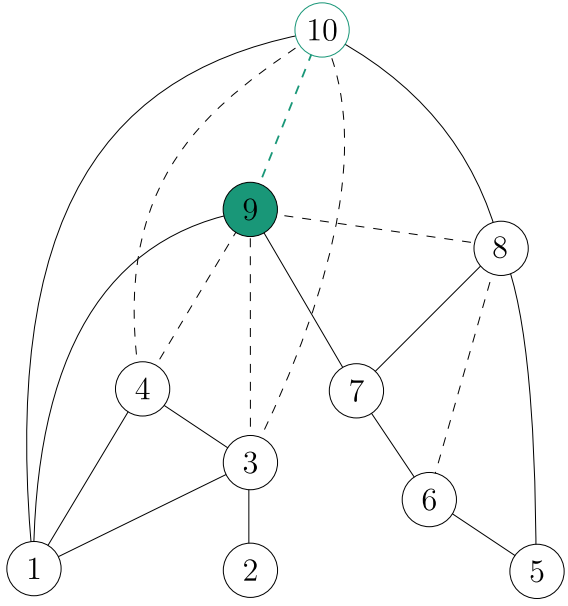
Copy the remaining edges to the parent if it is possible



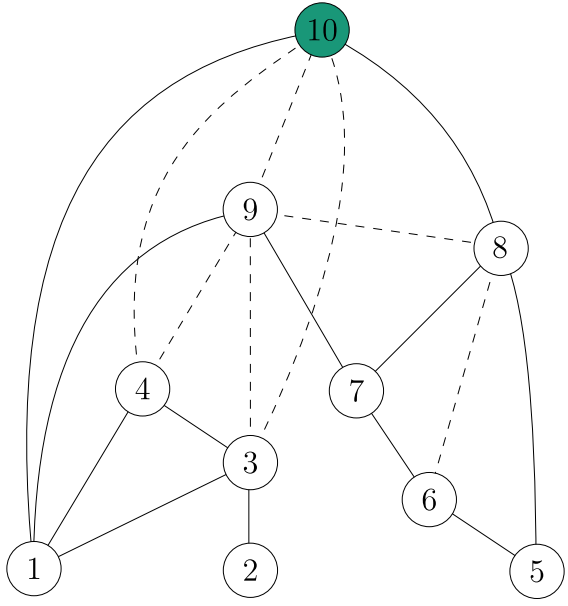
Contract vertex 9



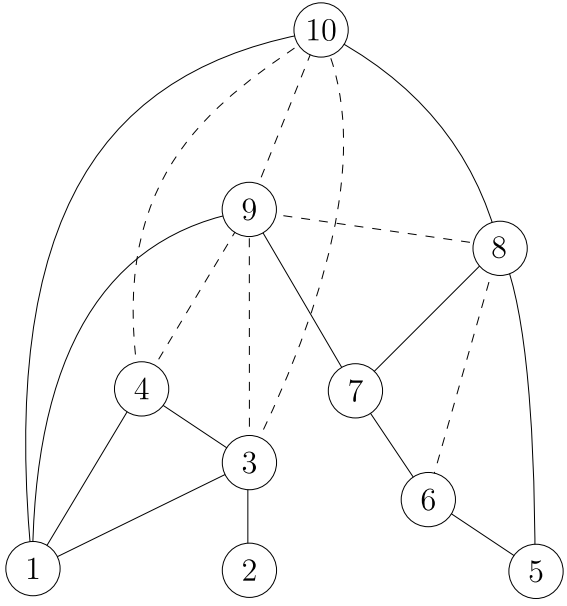
Find the outgoing edges



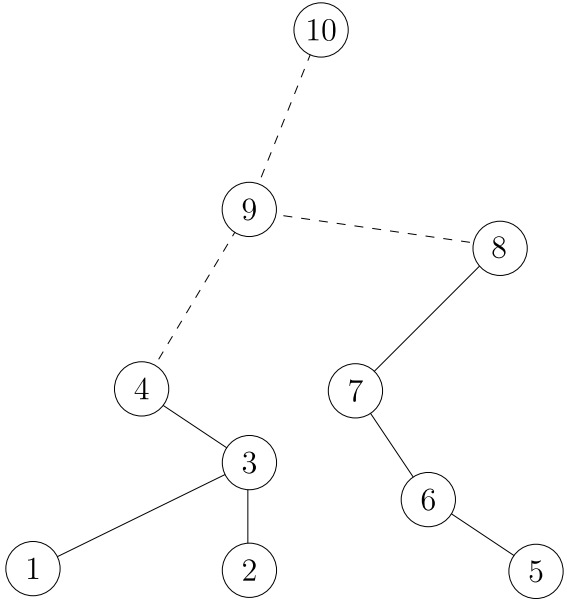
Set the smallest one as the parent



Contract vertex 10

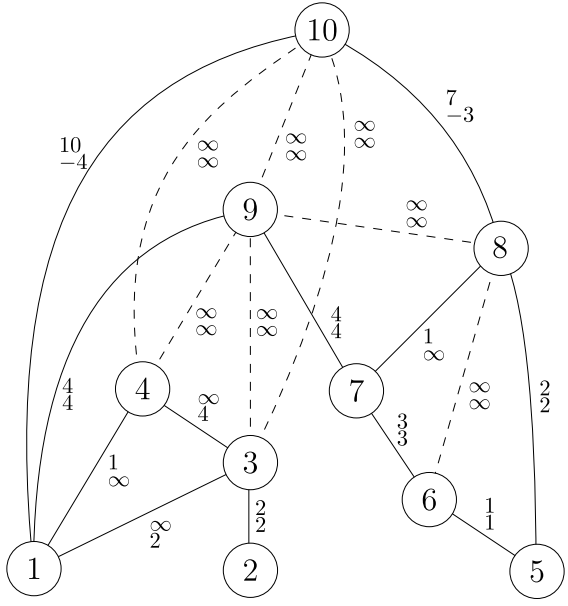


Contraction done

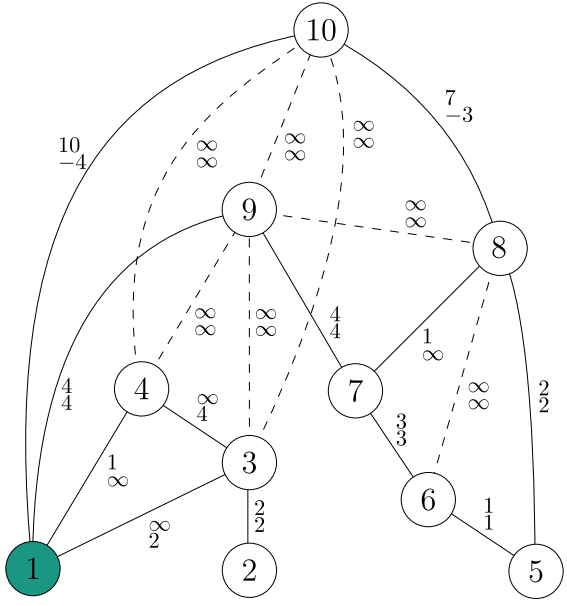


Elimination Tree

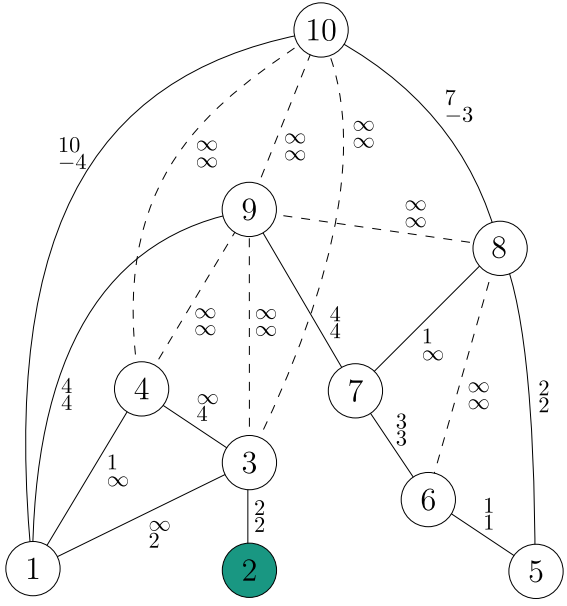
Basic Customization



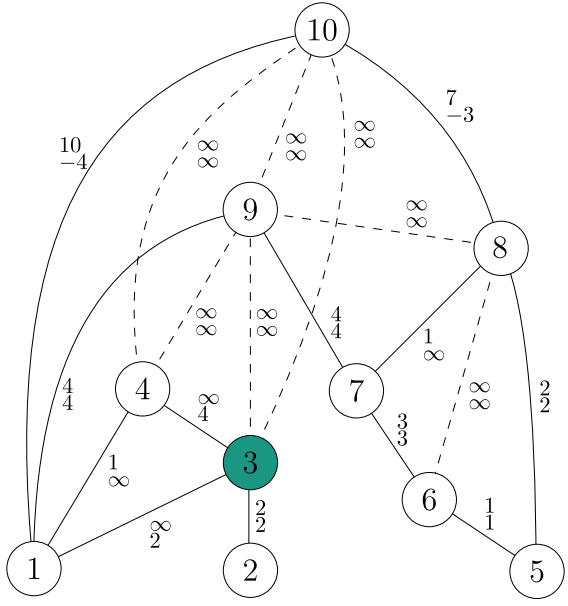
Apply the weights. Upweight is number above. downweight is number below.



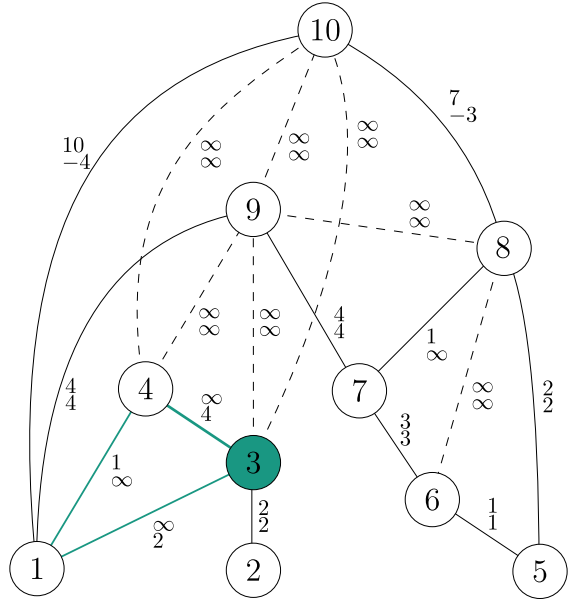
Customize for vertex 1



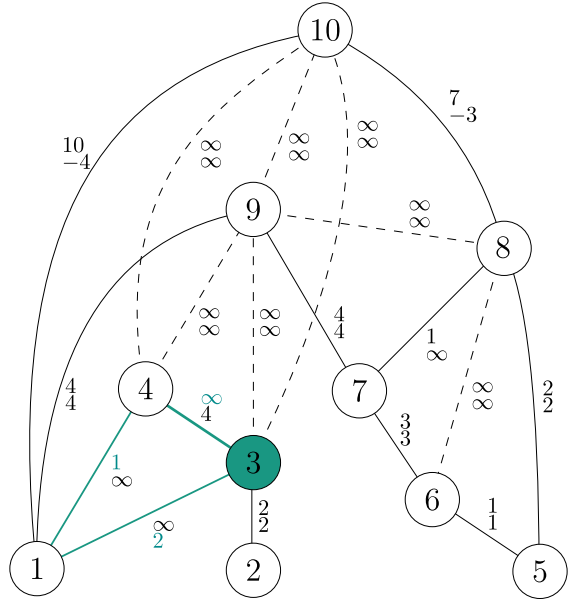
Customize for vertex 2



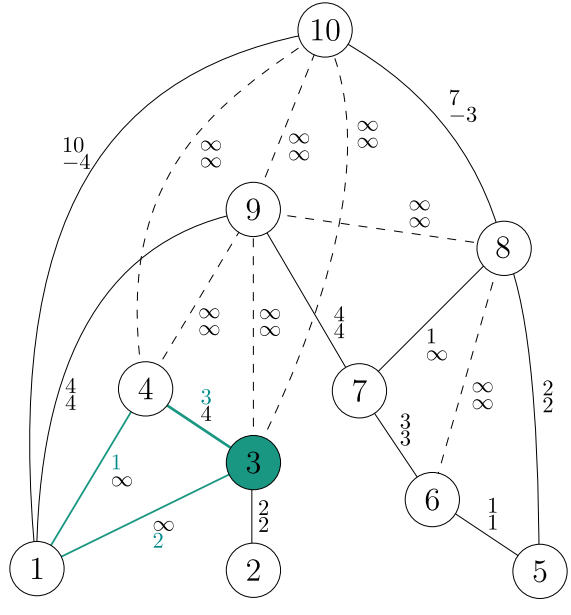
Customize for vertex 3

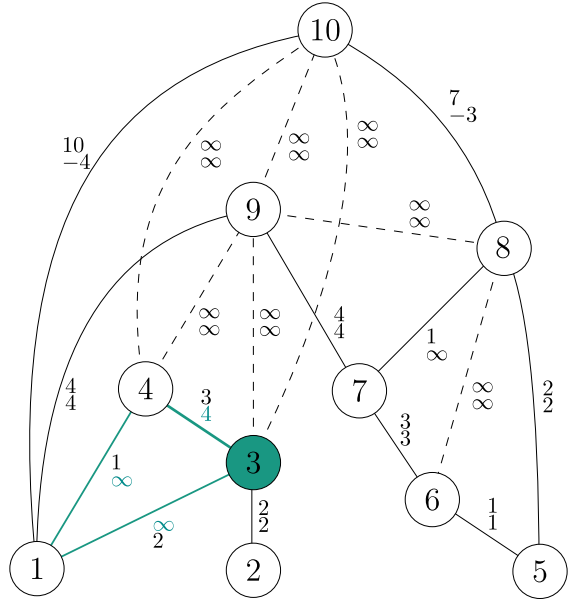


$\{1, 3, 4\}$ is a lower triangle of $(3, 4)$

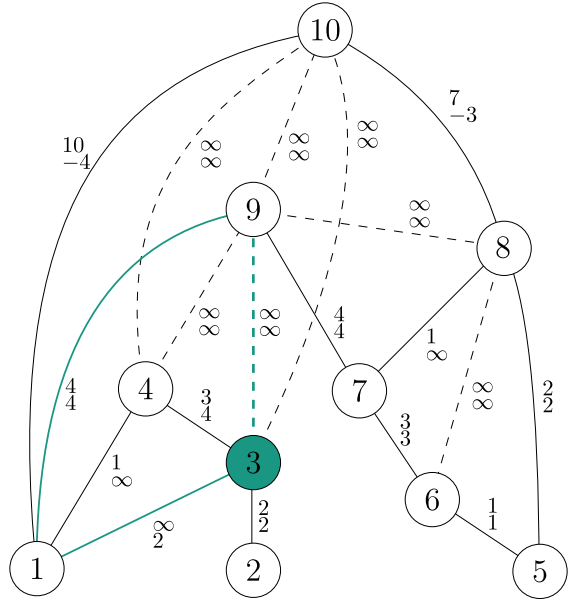


Update upweight if possible

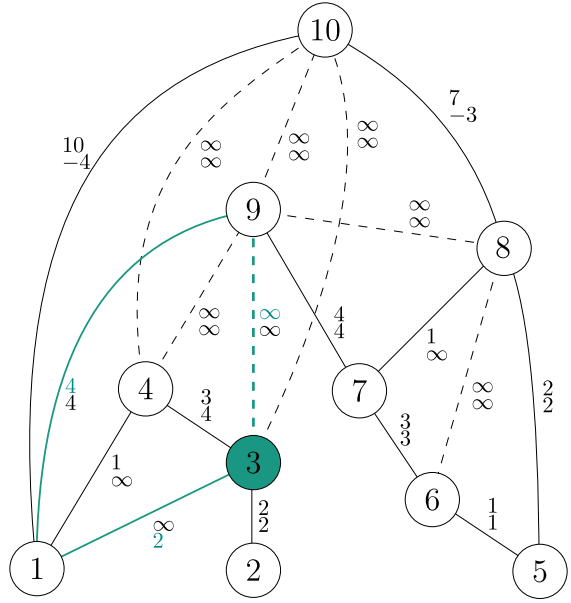

10
9
8
7
4
3
6
5
1
2
10
−4
7
−3
∞
∞
4
4
3
4
1
∞
2
2
3
3
1
1
∞
2



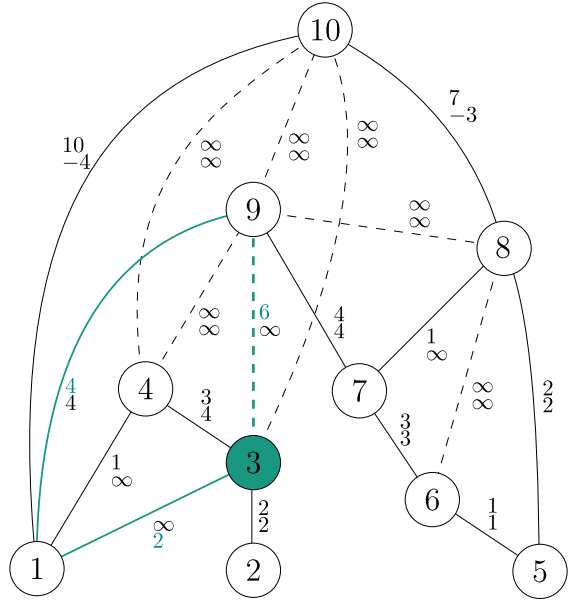
Update downweight if possible

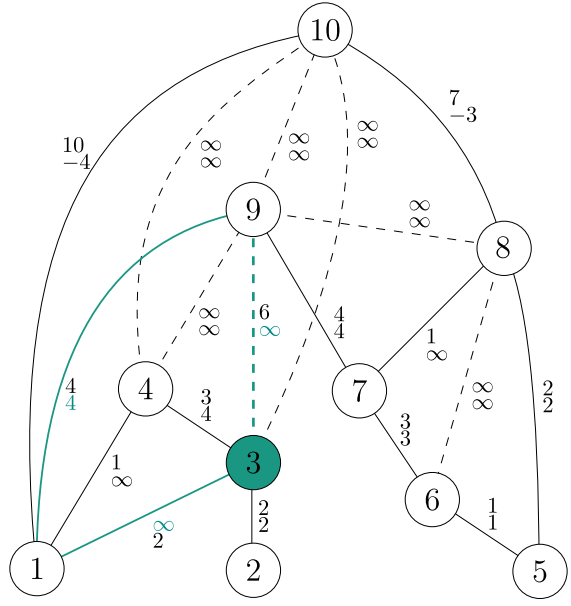


$\{1, 3, 9\}$ is a lower triangle of $(3, 9)$

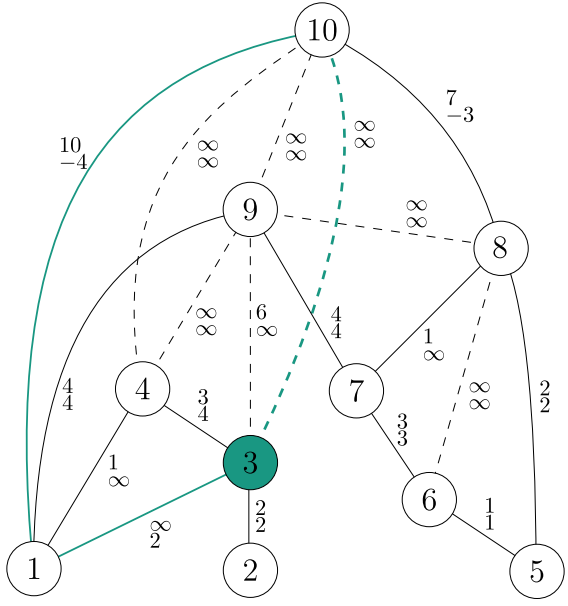


Update upweight if possible

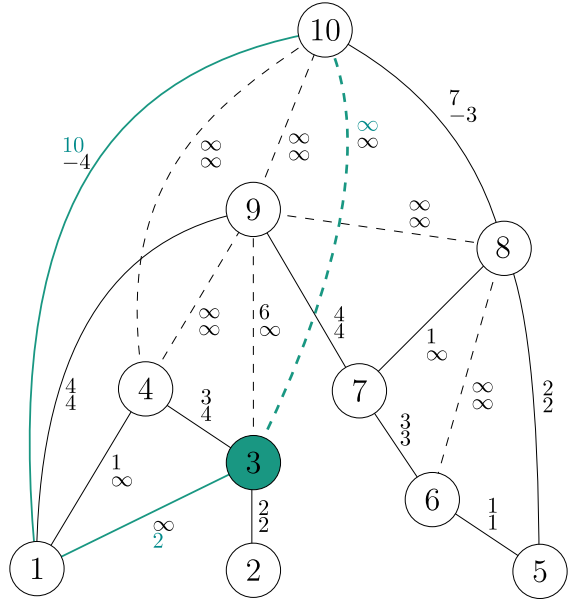

10
9
8
7
4
3
6
1
2
5
10
−4
7
−3
∞
∞
∞
∞
∞
∞
∞
∞
6
∞
4
4
1
∞
4
4
3
4
1
∞
∞
∞
2
2
3
3
1
∞
∞
2
2
1
1



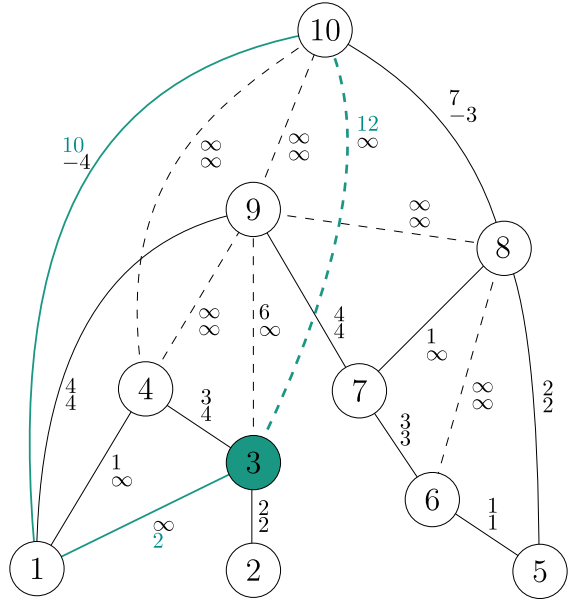
Update downweight if possible

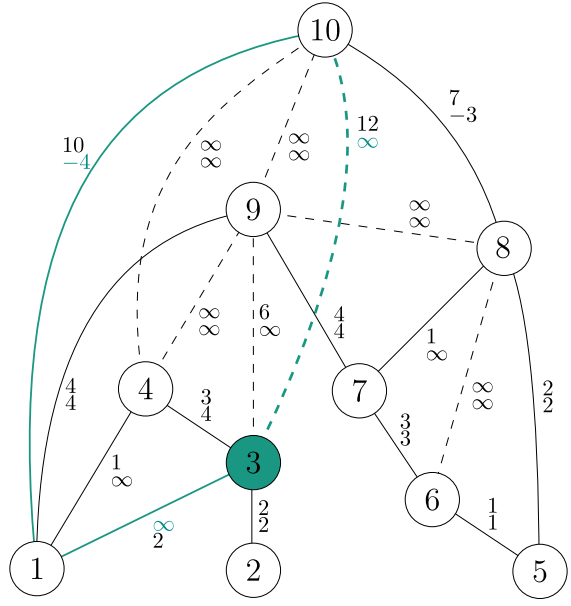


$\{1, 3, 10\}$ is a lower triangle of $(3, 10)$

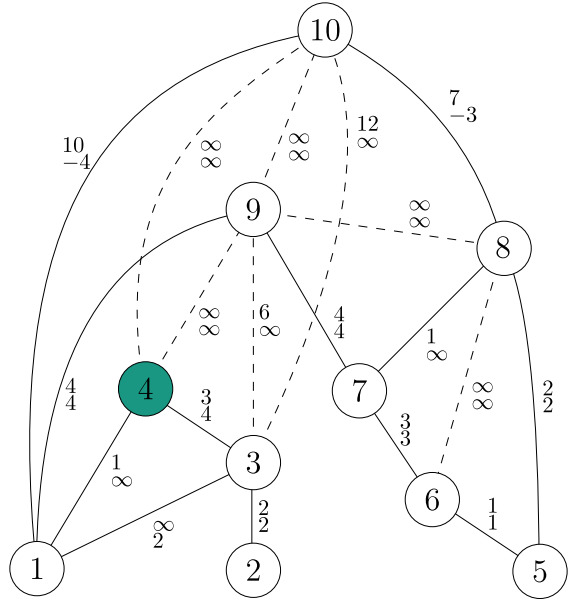


Update upweight if possible

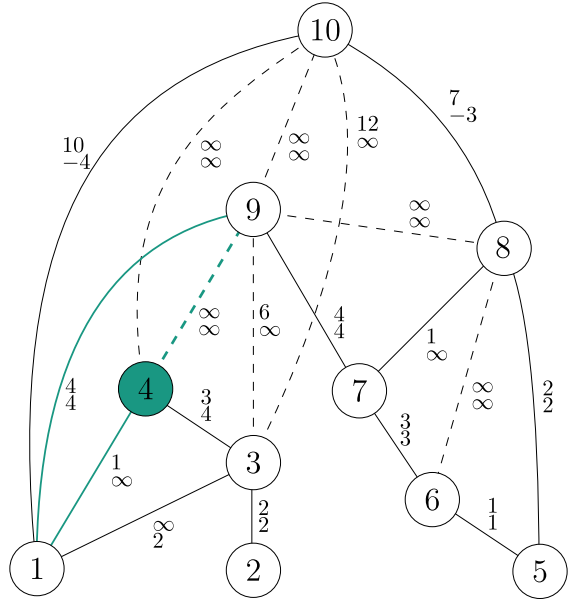

10
9
8
4
7
3
6
1
2
5
10
−4
7
−3
12
∞
∞
∞
∞
∞
∞
∞
∞
6
∞
4
4
1
∞
4
4
3
4
1
∞
∞
∞
2
2
3
3
1
∞
∞
2
2
1
1



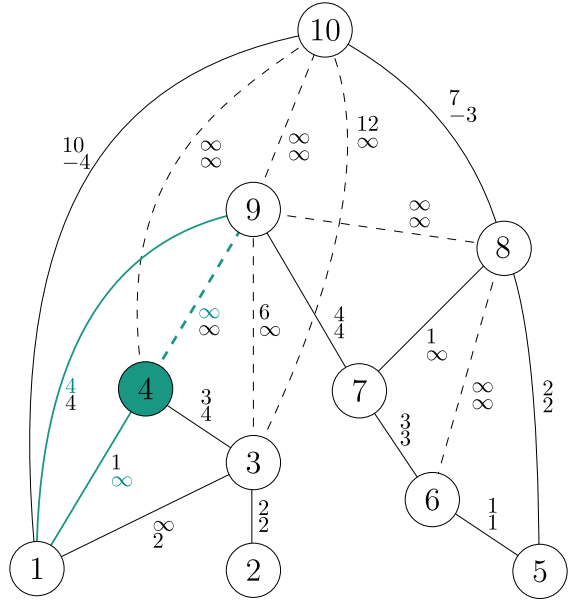
Update downweight if possible



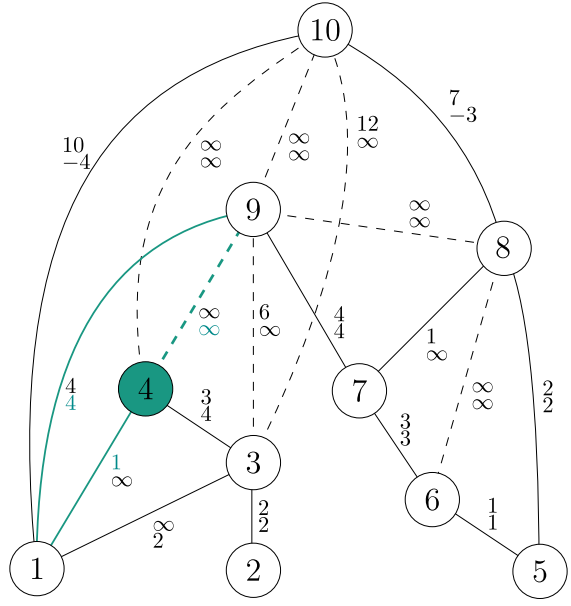
Customize for vertex 4



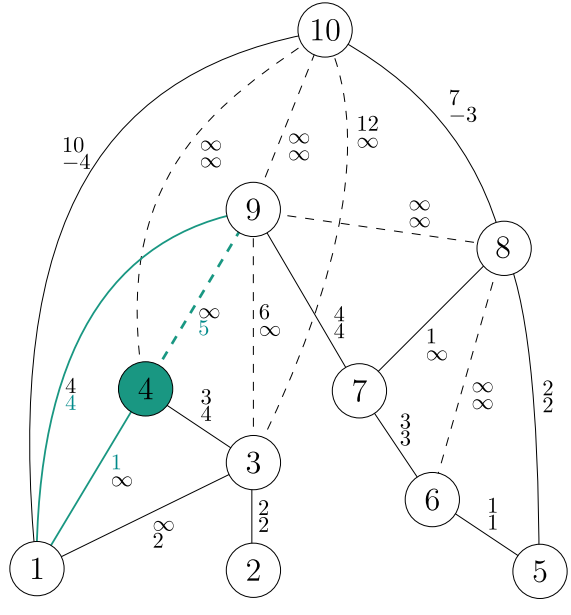
$\{1, 4, 9\}$ is a lower triangle of $(4, 9)$

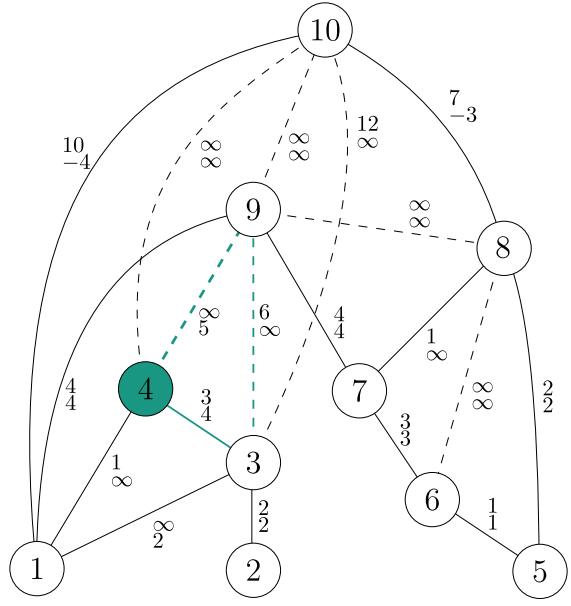


Update upweight if possible

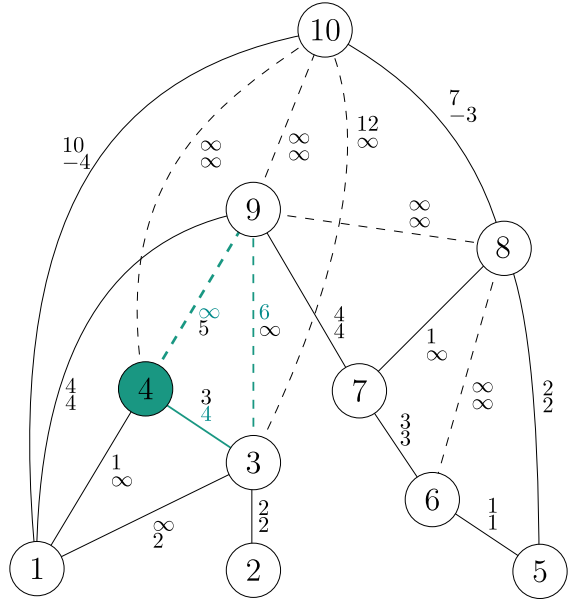


Update downweight if possible

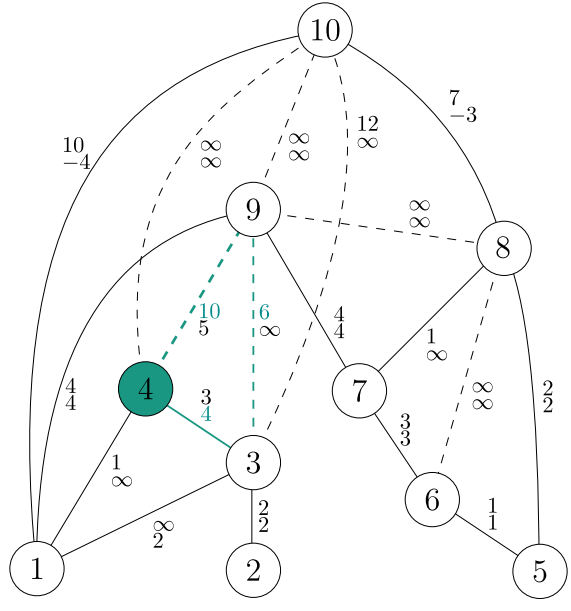

10
9
8
7
4
3
6
5
2
1
10
−4
7
−3
12
∞
∞
∞
∞
∞
∞
∞
∞
∞
5
6
∞
4
4
1
∞
4
4
3
4
∞
∞
2
2
1
∞
3
3
1
1
∞
2
2
2

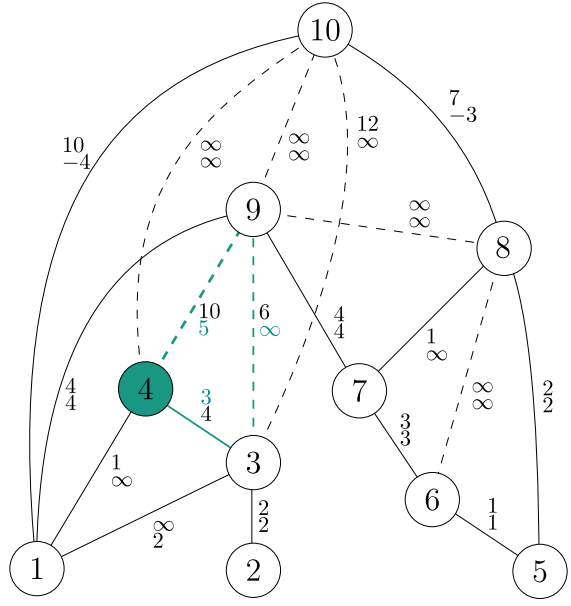


$\{3, 4, 9\}$ is a lower triangle of $(4, 9)$

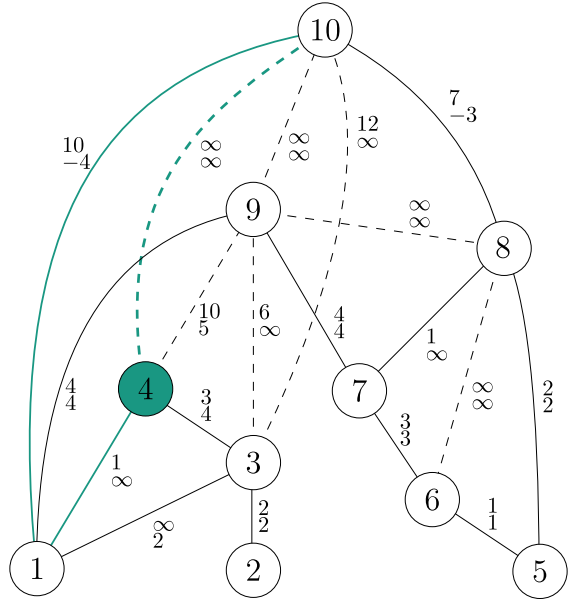


Update upweight if possible

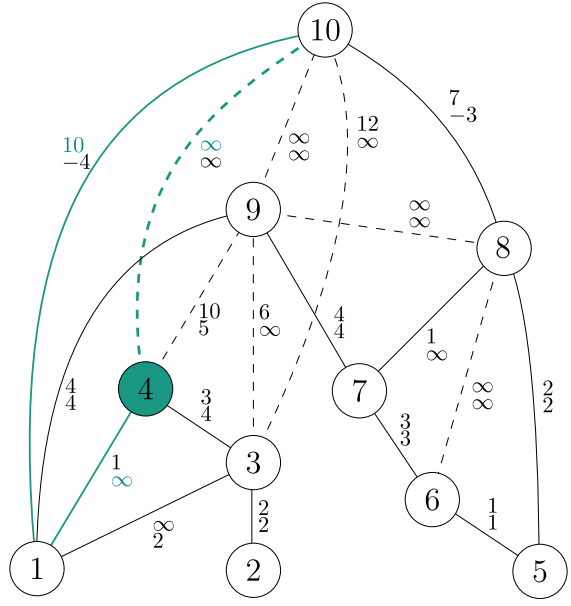

10
9
8
4
7
3
6
1
2
5
10
−4
∞
∞
∞
∞
12
∞
7
−3
∞
∞
10
5
6
∞
4
4
1
∞
4
4
3
4
∞
∞
2
2
3
3
1
∞
∞
2
2
2
1
1



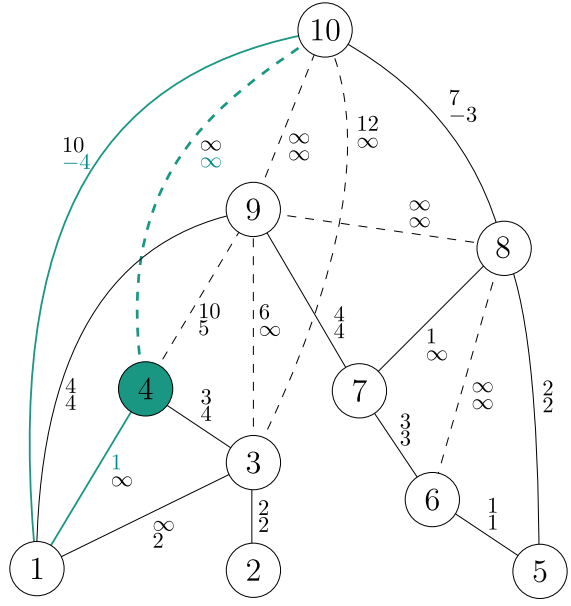
Update downweight if possible



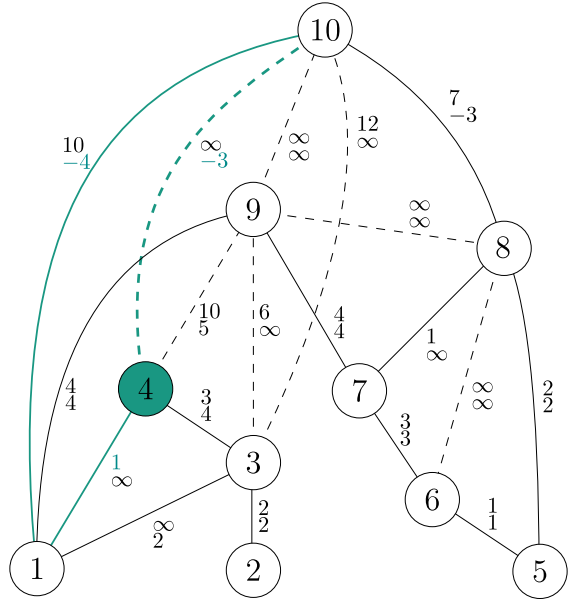
$\{1, 4, 10\}$ is a lower triangle of $(4, 10)$

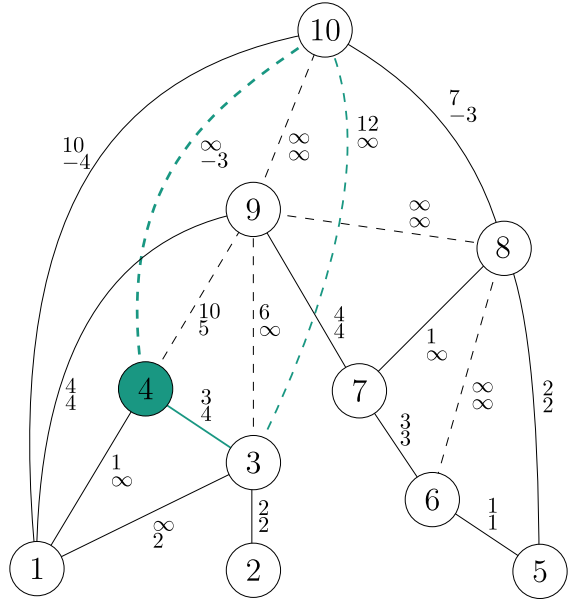


Update upweight if possible

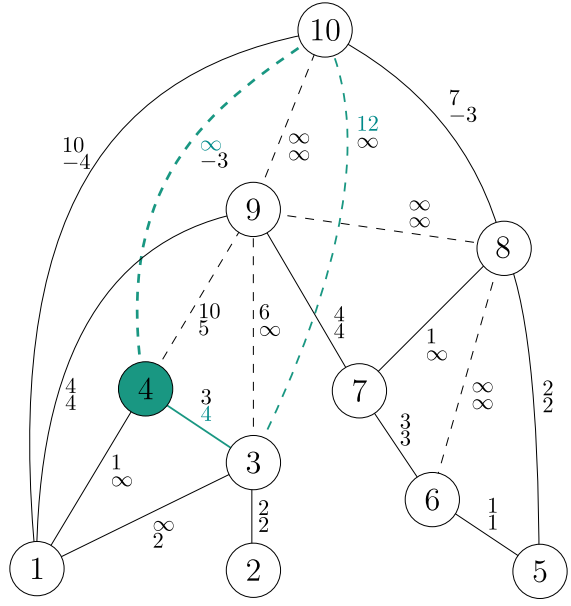


Update downweight if possible

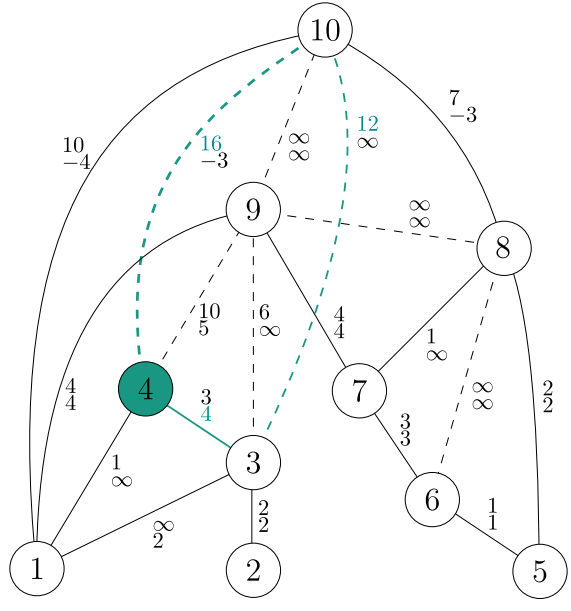

10
9
8
7
4
3
6
5
2
1
10
−4
∞
−3
∞
∞
12
∞
7
−3
∞
∞
10
5
6
∞
4
4
1
∞
4
4
3
4
∞
∞
2
2
3
3
1
∞
∞
2
2
2
1
1

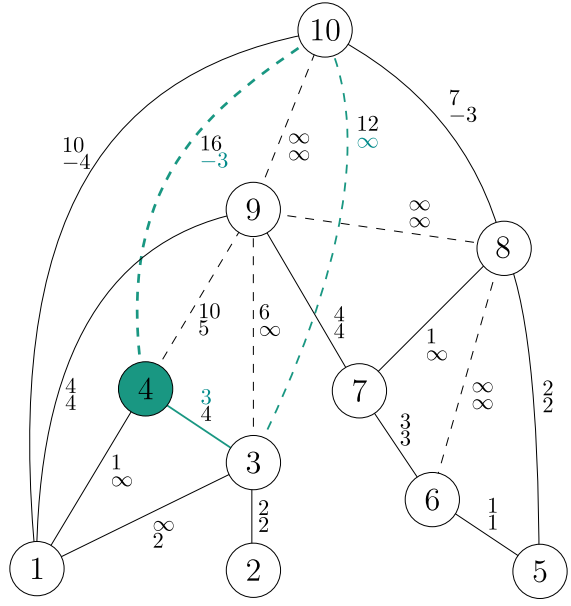


$\{3, 4, 10\}$ is a lower triangle of $(4, 10)$

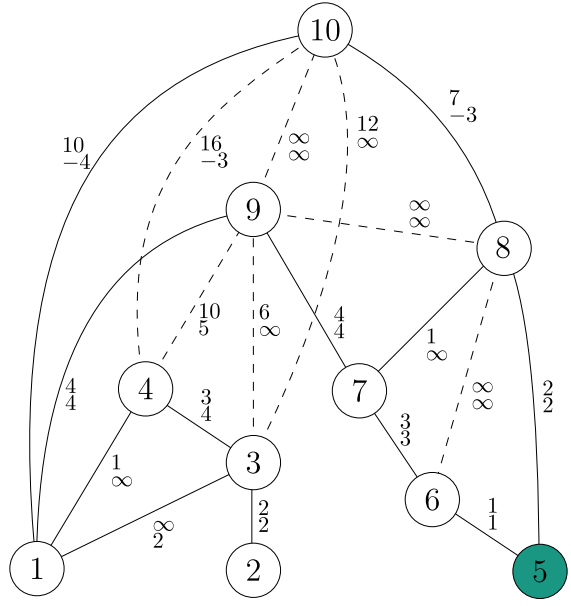


Update upweight if possible

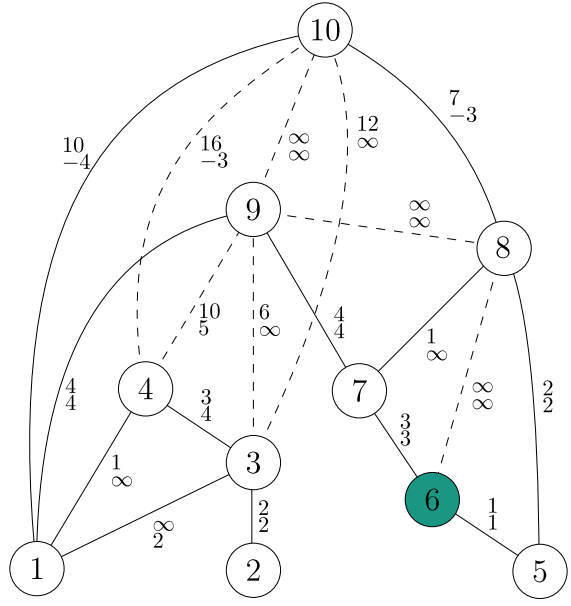

10
9
8
7
4
3
6
1
2
5
10
−4
16
−3
∞
∞
12
∞
7
−3
∞
∞
10
5
6
∞
4
4
1
∞
4
4
3
4
∞
∞
2
2
3
3
1
∞
∞
2
2
2
1
1



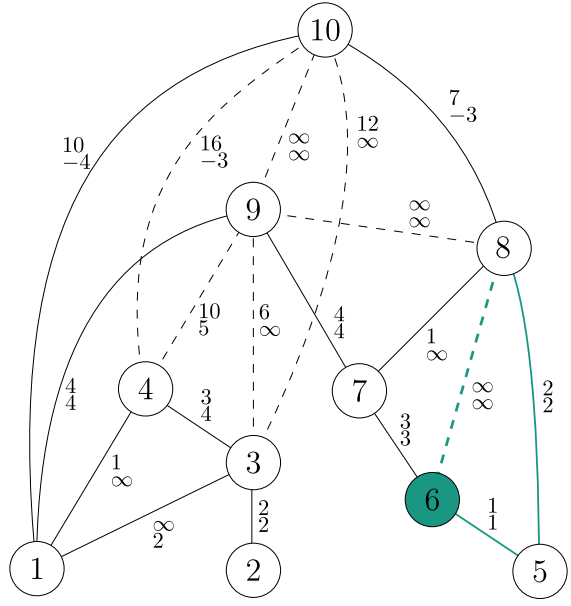
Update downweight if possible



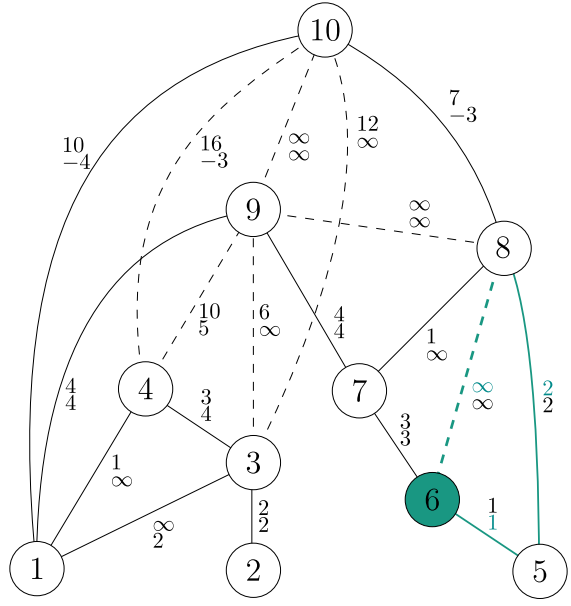
Customize for vertex 5



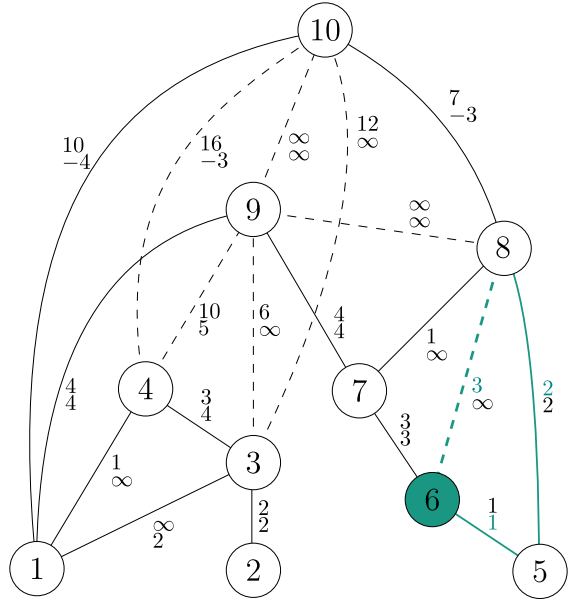
Customize for vertex 6

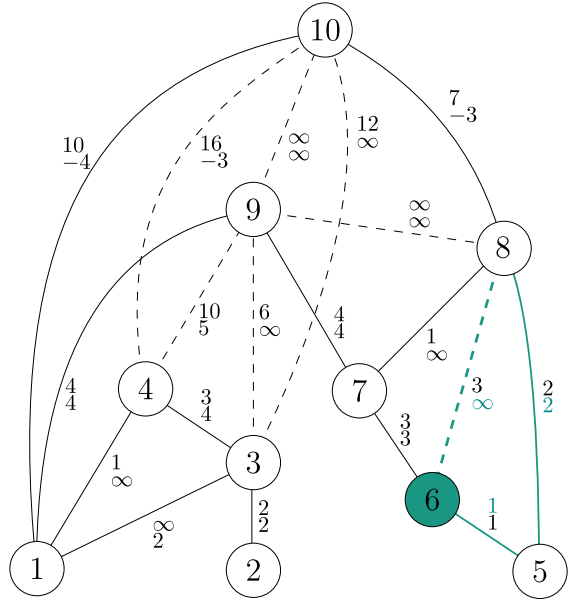


$\{5, 6, 8\}$ is a lower triangle of $(6, 8)$

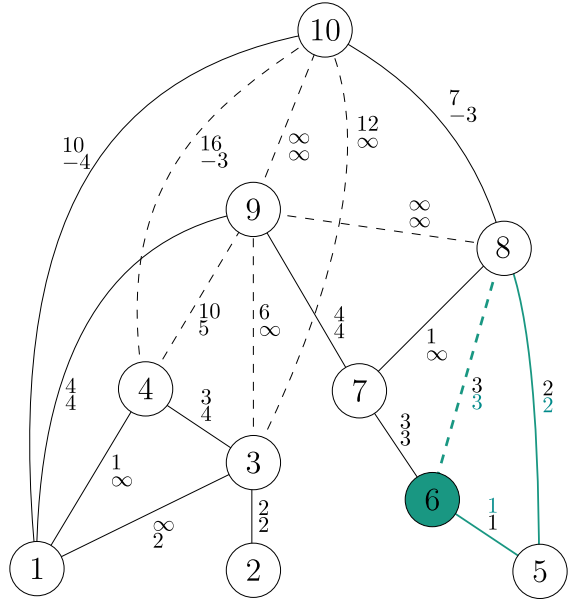


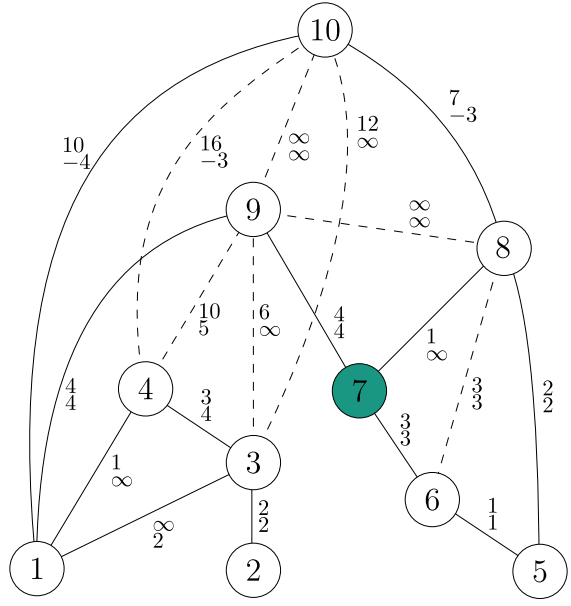
Update upweight if possible


10
9
8
7
4
3
6
5
1
2
10
−4
16
−3
∞
∞
12
∞
7
−3
∞
∞
10
5
6
∞
4
4
1
∞
4
4
3
4
3
∞
2
2
1
∞
3
3
∞
2
1
1

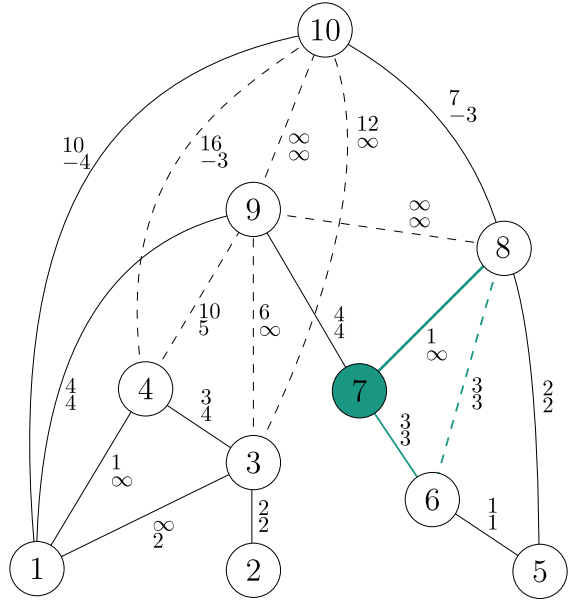


Update downweight if possible

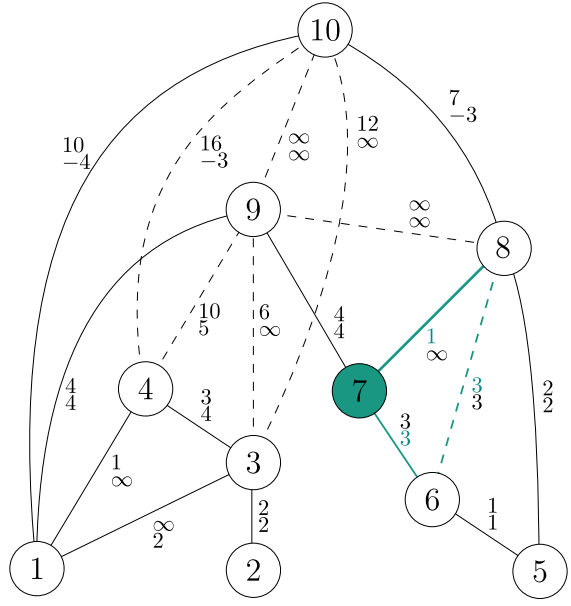

10
9
8
4
7
3
6
1
2
5
10
−4
16
−3
∞
∞
12
∞
7
−3
∞
∞
10
5
6
∞
4
4
1
∞
4
4
3
4
3
3
3
3
2
2
1
∞
∞
2
2
2
1
1



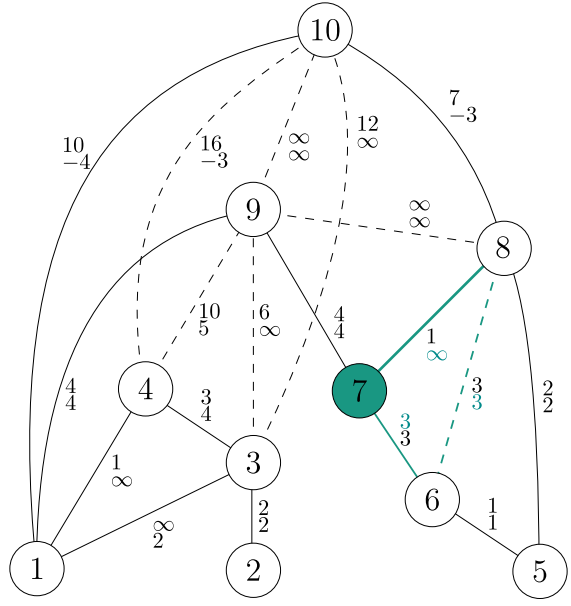
Customize for vertex 7



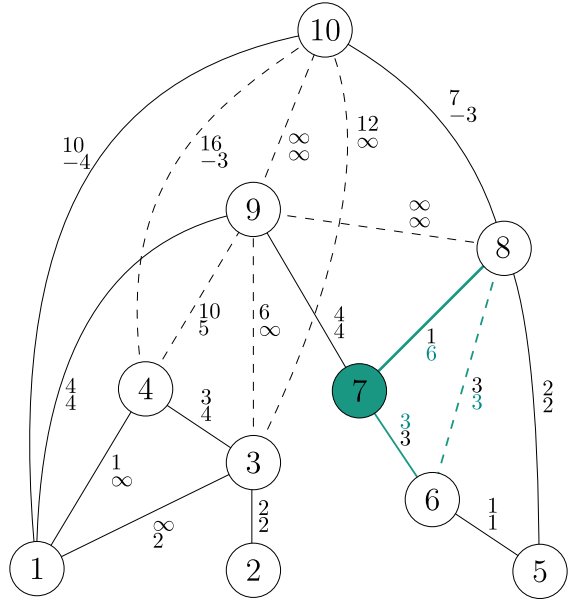
$\{6, 7, 8\}$ is a lower triangle of $(7, 8)$

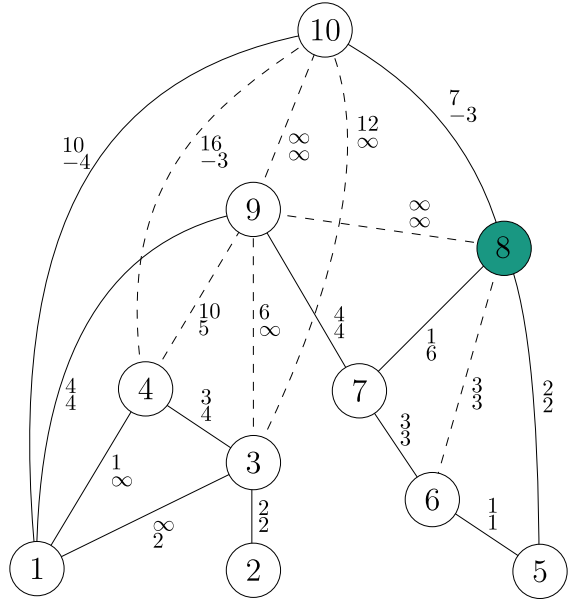


Update upweight if possible

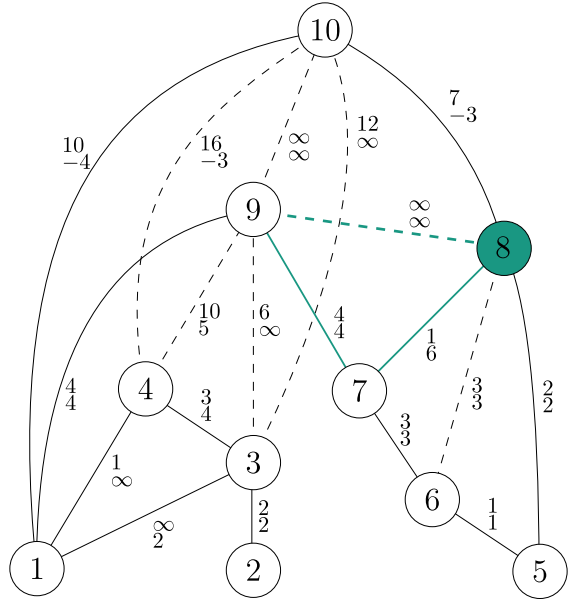


Update downweight if possible

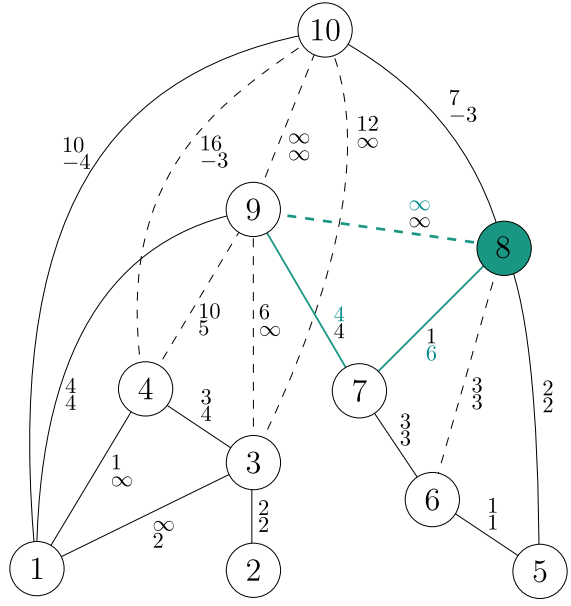

10
9
8
7
4
3
6
5
1
2
10
−4
16
−3
∞
∞
12
∞
7
−3
∞
∞
10
5
6
∞
4
4
1
6
4
4
3
4
3
3
3
3
2
2
1
∞
∞
2
2
2
1
1



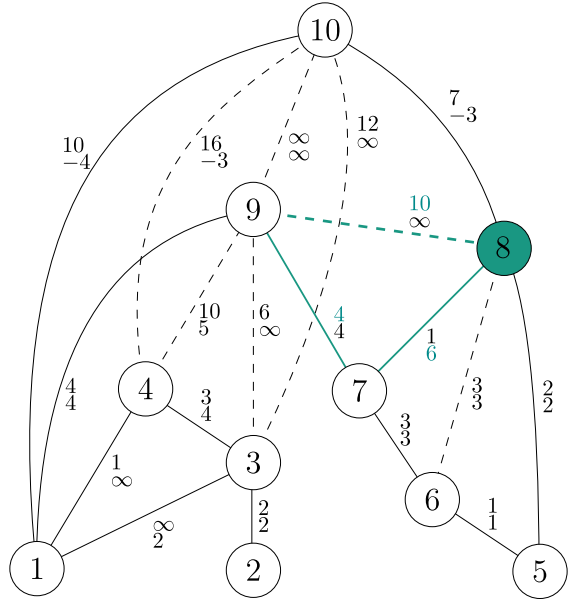
Customize for vertex 8

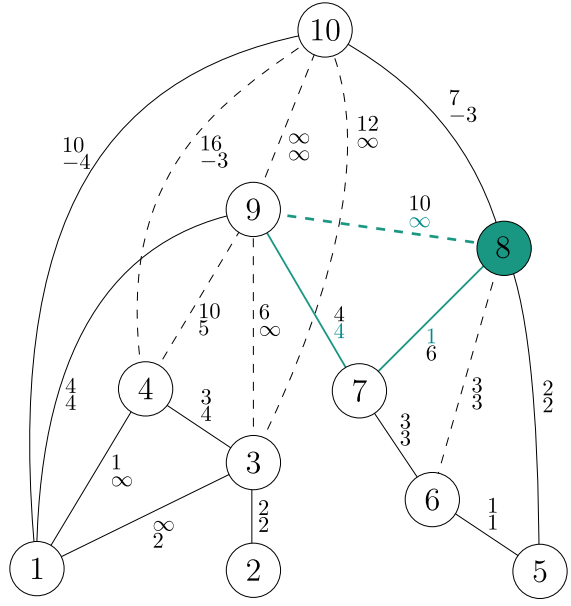


$\{7, 8, 9\}$ is a lower triangle of $(8, 9)$

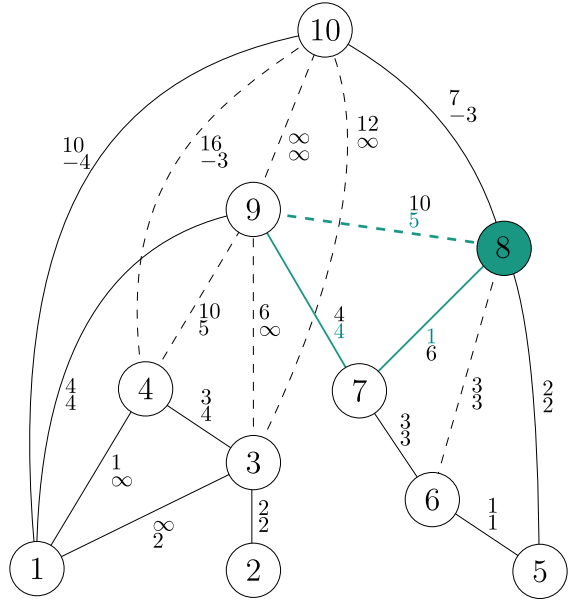


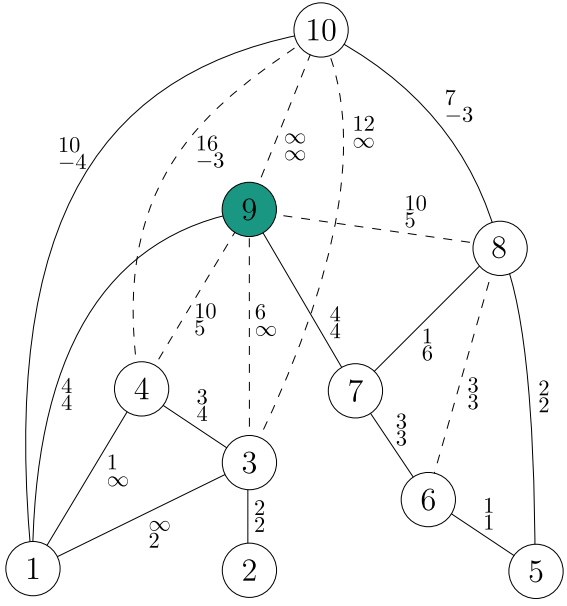
Update upweight if possible


10
9
8
7
4
3
6
5
2
1
10
−4
16
−3
∞
∞
12
∞
7
−3
10
∞
10
5
6
∞
4
4
1
6
4
4
3
4
3
3
3
3
2
2
1
∞
1
∞
2
2
2
1
1

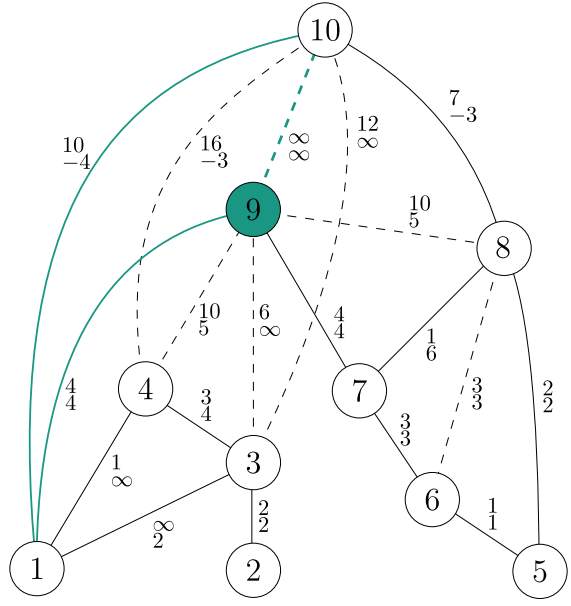


Update downweight if possible

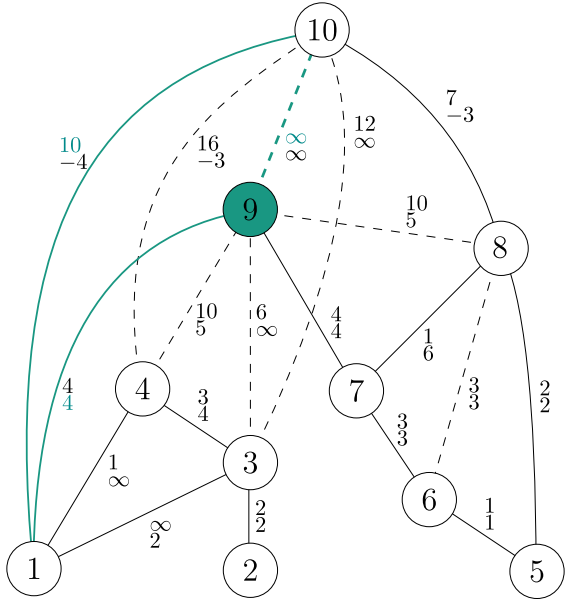

10
9
8
7
4
3
6
5
1
2
10
−4
16
−3
∞
∞
12
∞
7
−3
10
5
4
4
10
5
6
∞
4
4
1
6
4
4
3
4
3
3
3
3
2
2
1
∞
2
∞
2
2
1
1



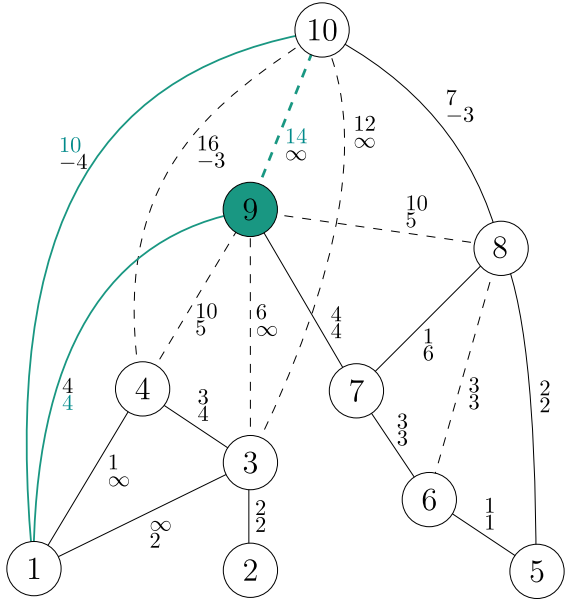
Customize for vertex 9

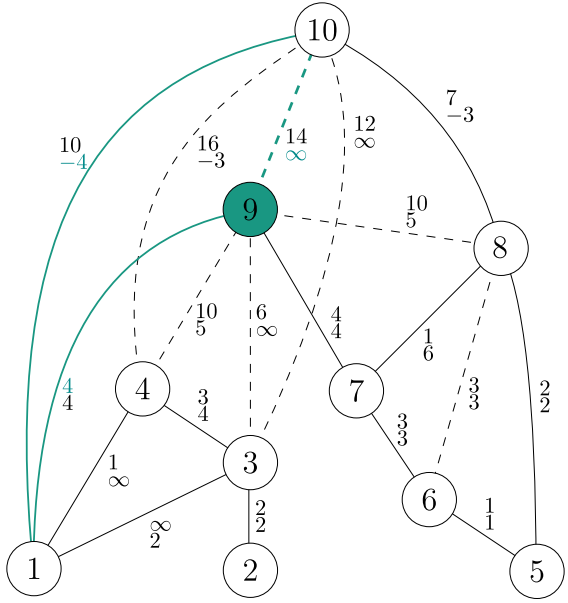


$\{1, 9, 10\}$ is a lower triangle of $(9, 10)$

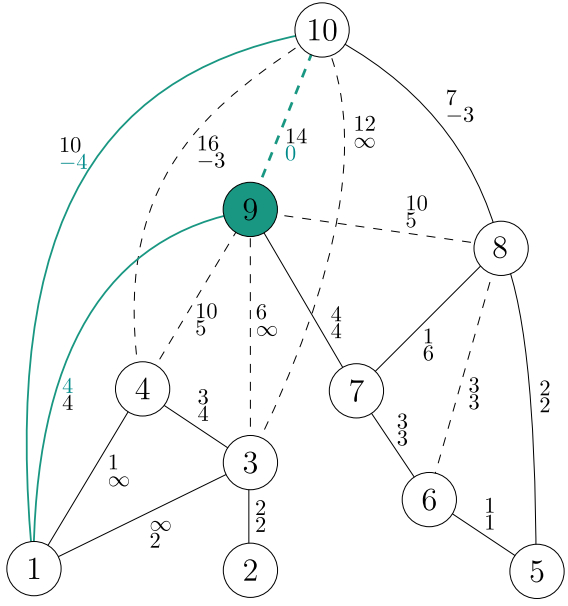


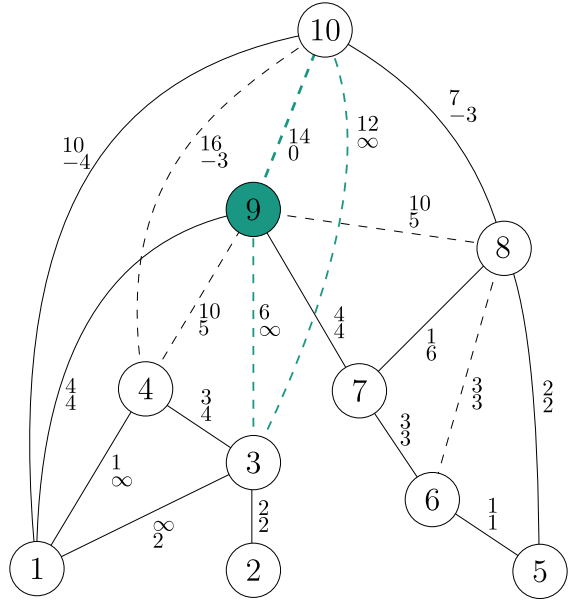
Update upweight if possible


10
9
8
7
4
3
6
5
1
2
10
−4
16
−3
14
∞
12
∞
7
−3
10
5
10
5
6
∞
4
4
1
6
4
4
3
4
3
3
3
3
2
2
1
∞
1
∞
2
2
1
1

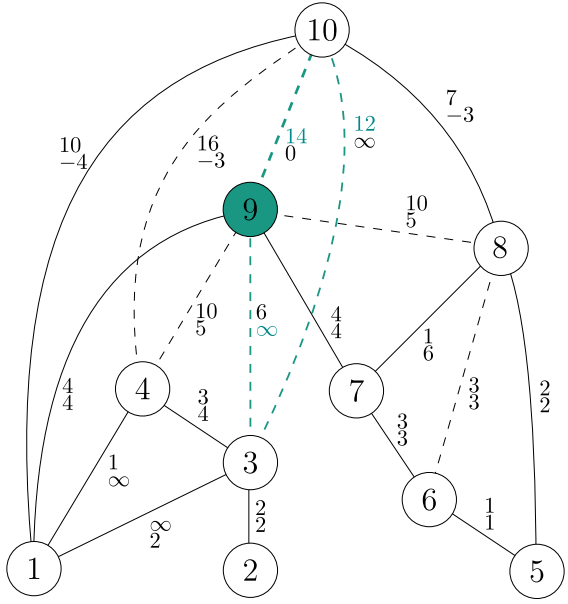


Update downweight if possible

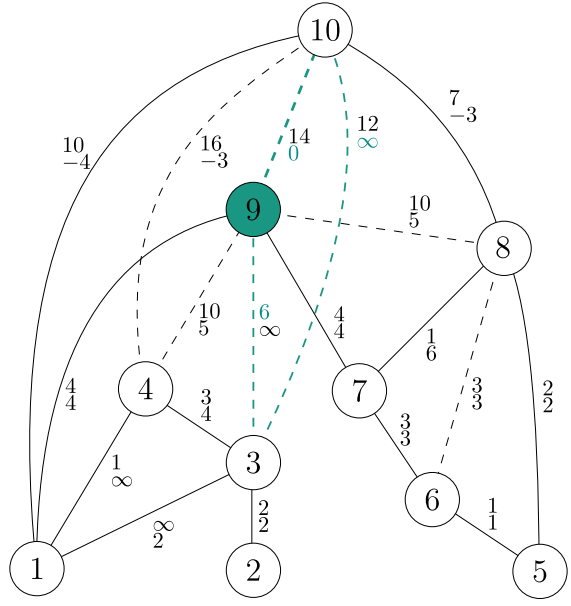

10
9
8
7
4
3
6
5
2
1
10
−4
16
−3
14
0
12
∞
7
−3
10
5
10
5
6
∞
4
4
1
6
4
4
3
4
3
3
3
3
2
2
1
∞
2
2
1
1
∞
2



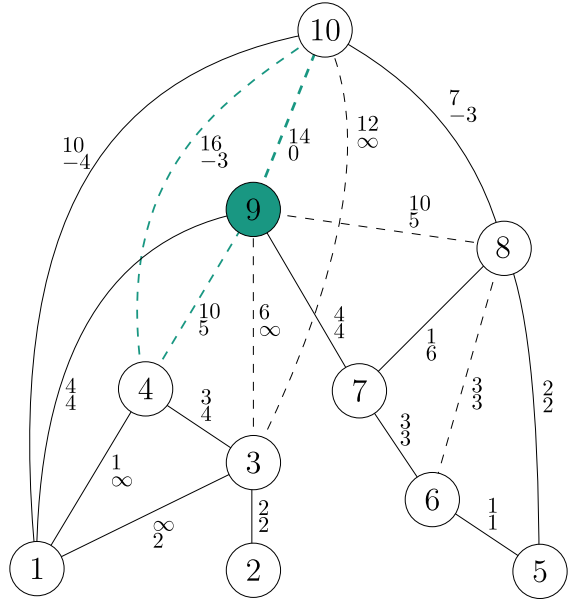
$\{3, 9, 10\}$ is a lower triangle of $(9, 10)$



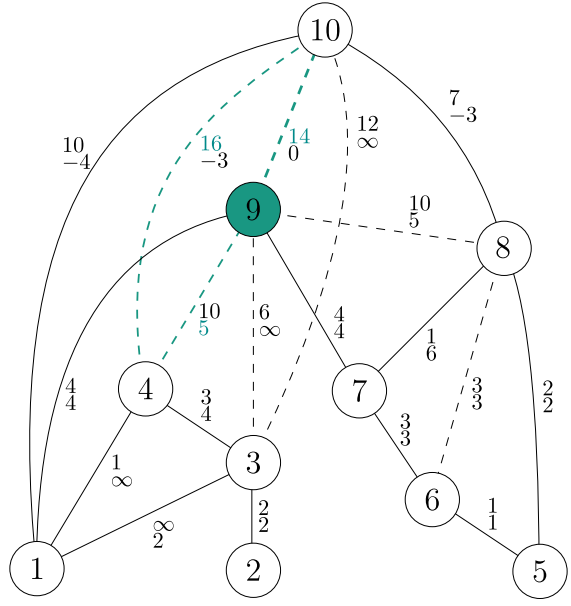
Update upweight if possible



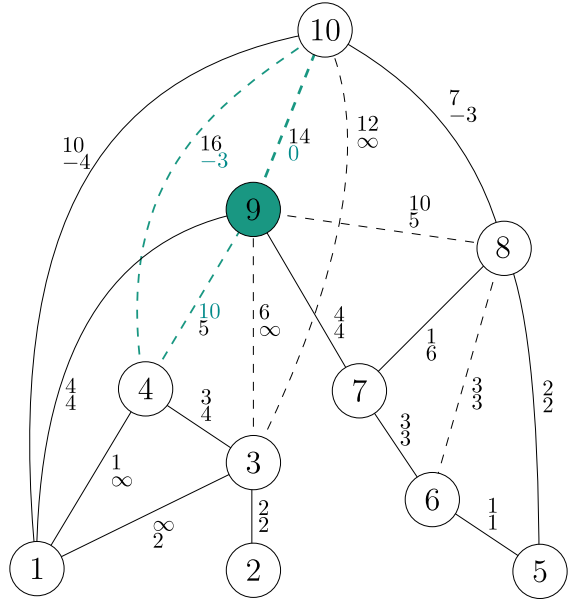
Update downweight if possible



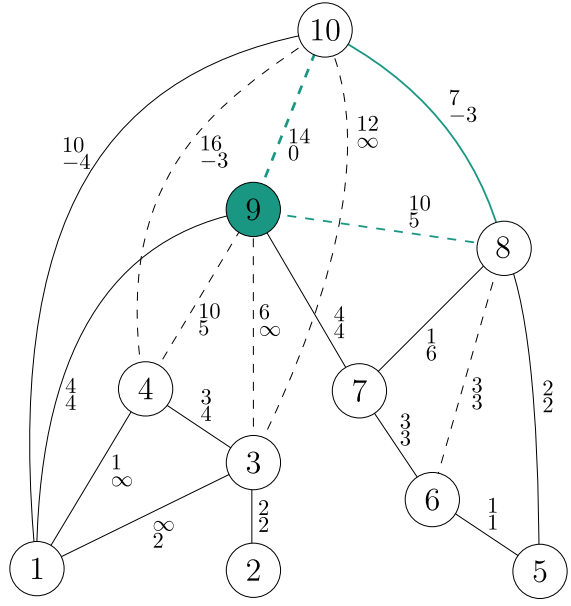
$\{4, 9, 10\}$ is a lower triangle of $(9, 10)$



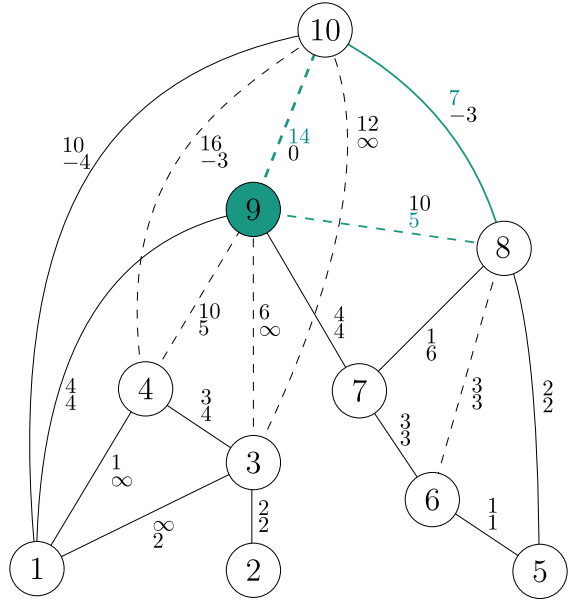
Update upweight if possible



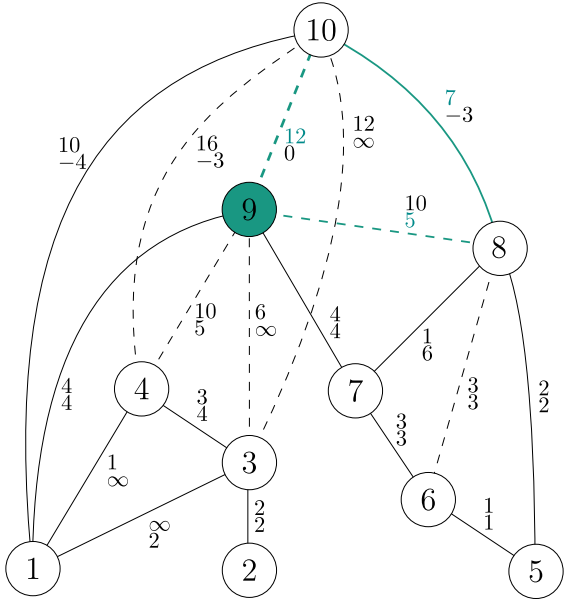
Update downweight if possible

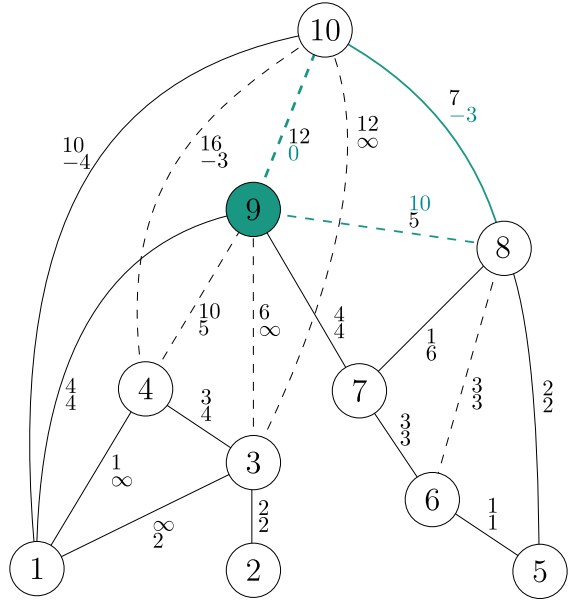


$\{8, 9, 10\}$ is a lower triangle of $(9, 10)$

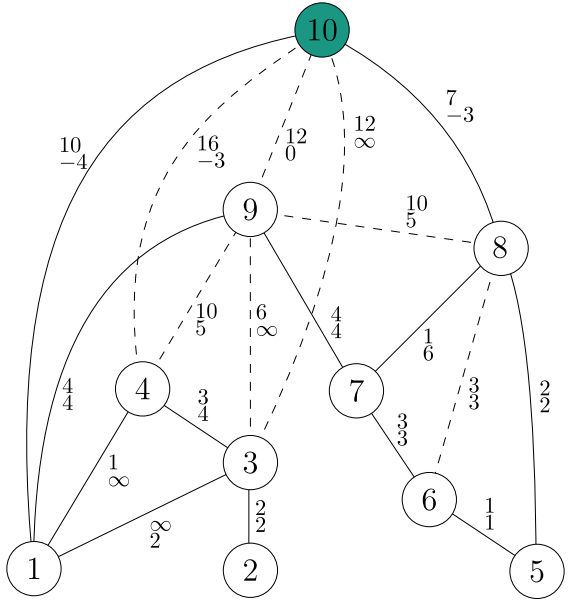


Update upweight if possible

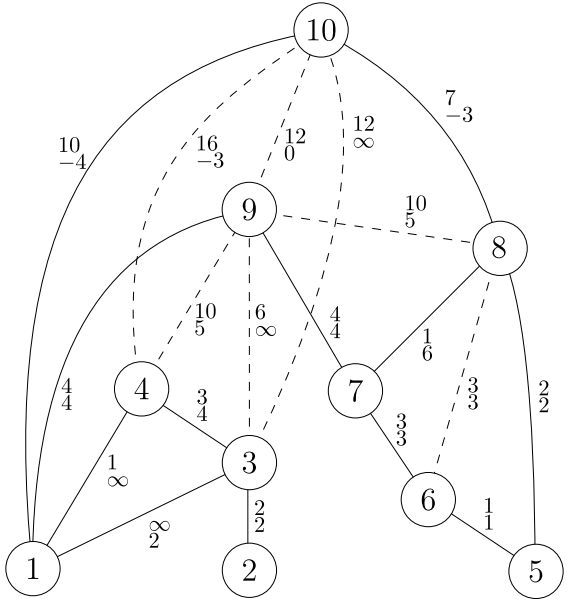

10
9
8
7
4
3
6
5
2
1
10
−4
16
−3
12
0
12
∞
7
−3
10
5
10
5
6
∞
4
4
1
6
4
4
3
4
3
3
3
3
2
2
1
∞
1
∞
2
2
1
1



Update downweight if possible

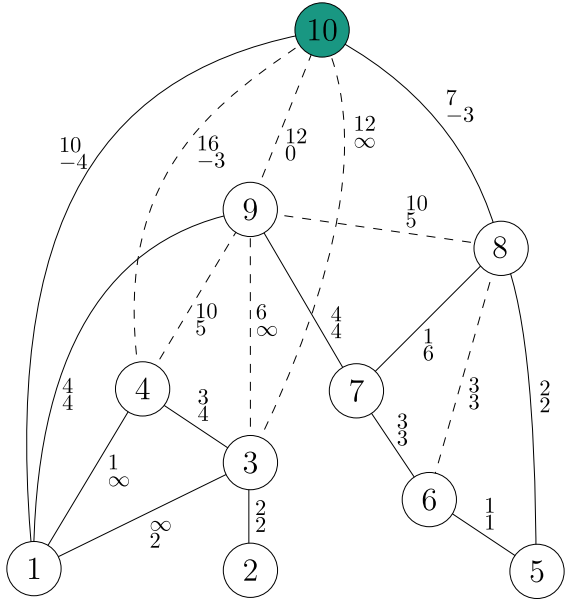


Customize for vertex 10

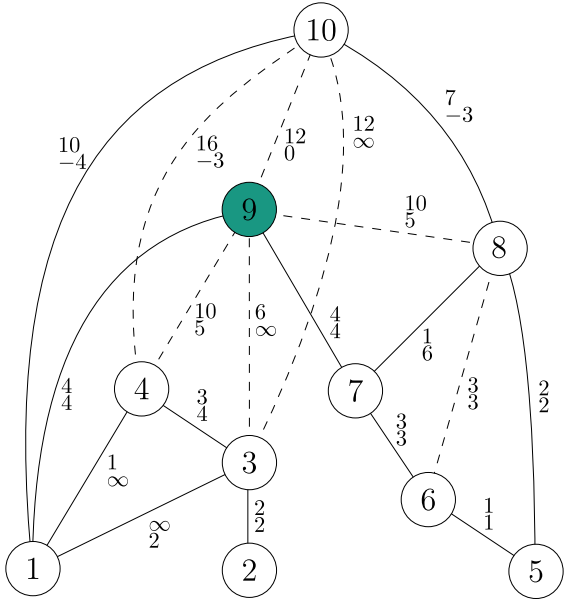


Basic customization done

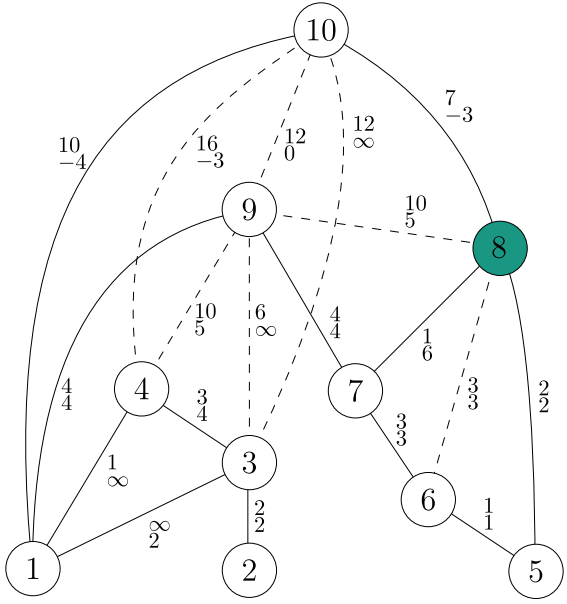
Perfect Customization



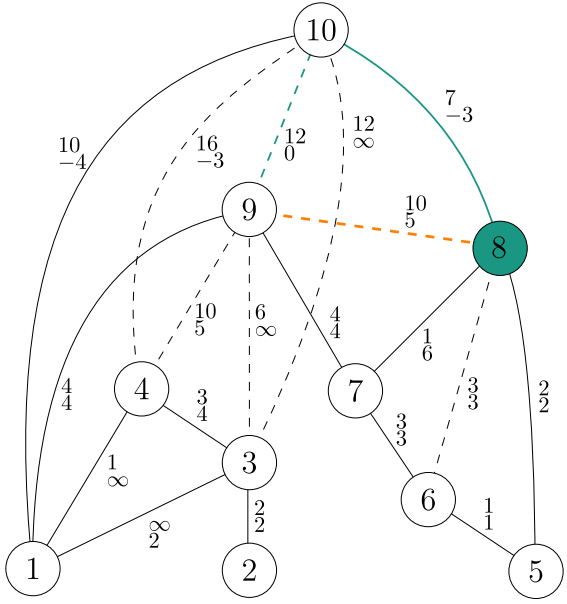
Customize for vertex 10



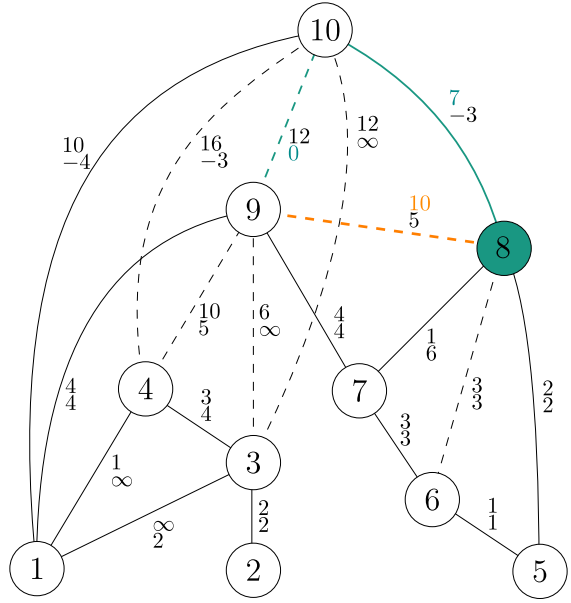
Customize for vertex 9



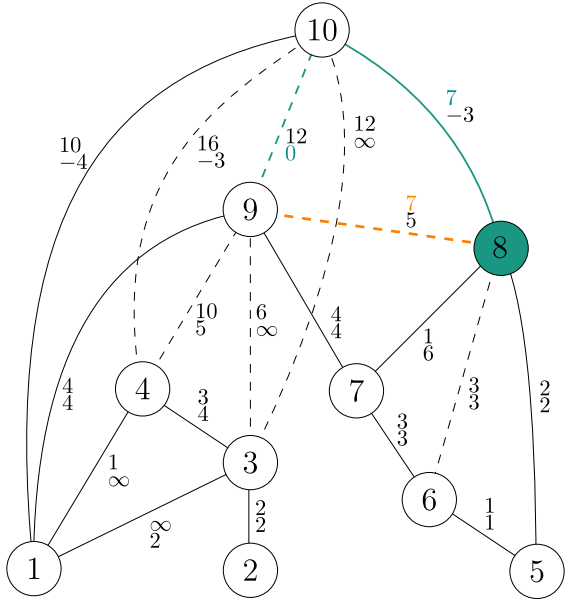
Customize for vertex 8

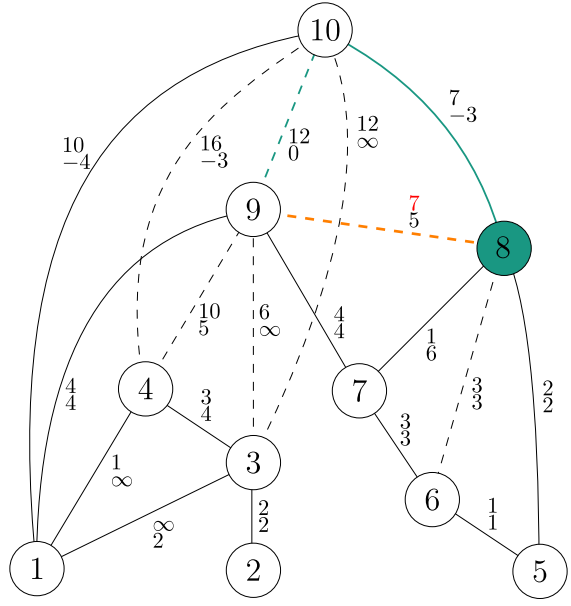


$\{8, 9, 10\}$ is a upper triangle of $(8, 9)$ and a middle triangle of $(8, 10)$

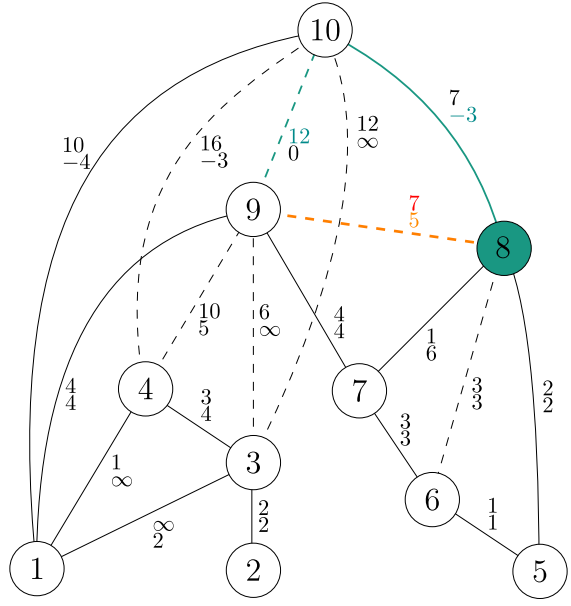


Update upweight for upper triangle if possible

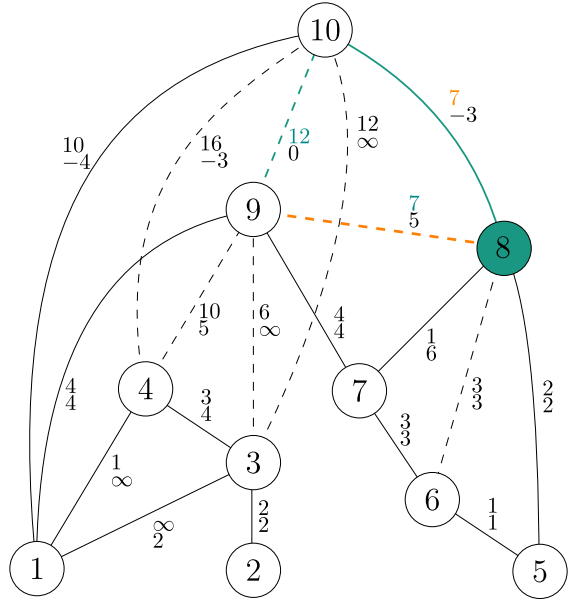

10
9
8
7
4
3
6
1
2
5
10
−4
16
−3
12
0
12
∞
7
−3
7
5
10
5
6
∞
4
4
1
6
4
4
3
4
3
3
3
3
2
2
1
∞
∞
2
2
2
1
1



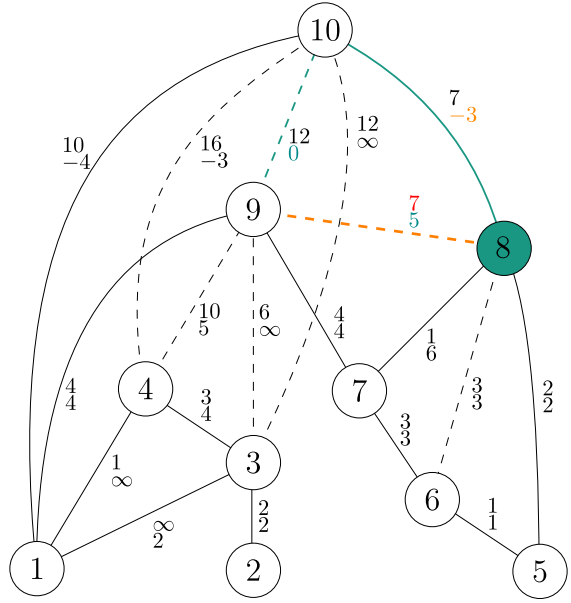
Mark edge as deleted because its weight was changed



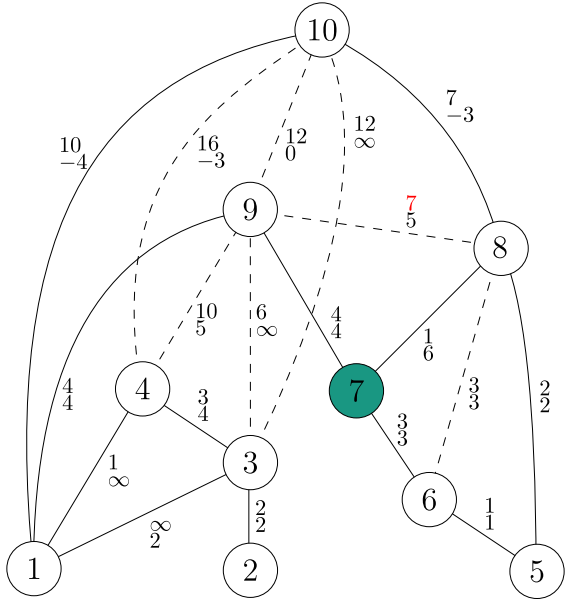
Update downweight for upper triangle if possible



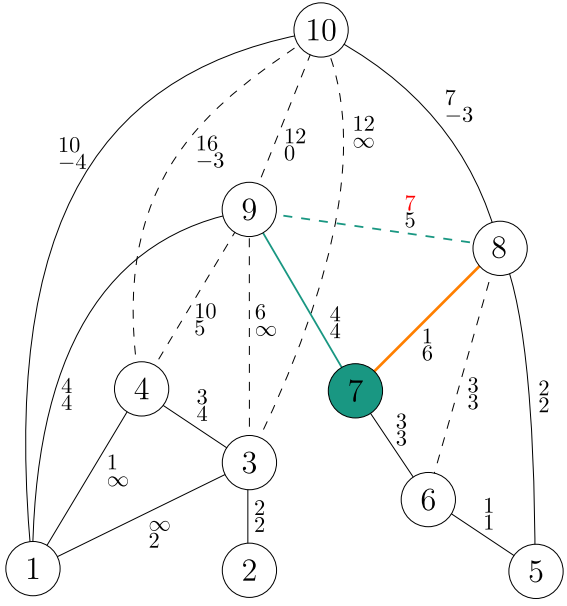
Update upweight for middle triangle if possible



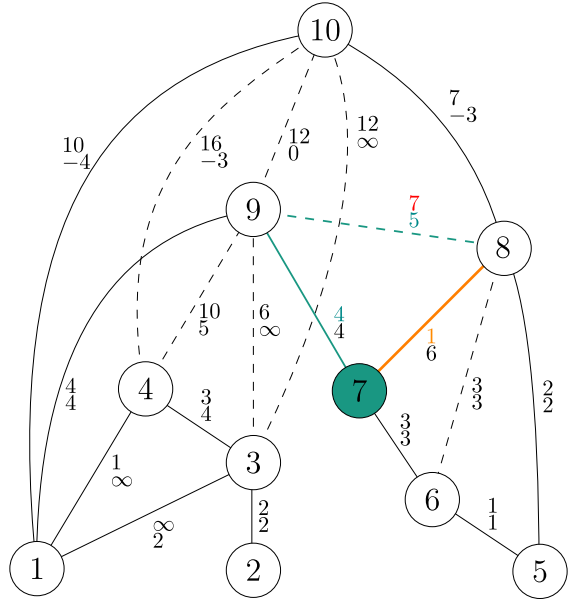
Update downweight for middle triangle if possible



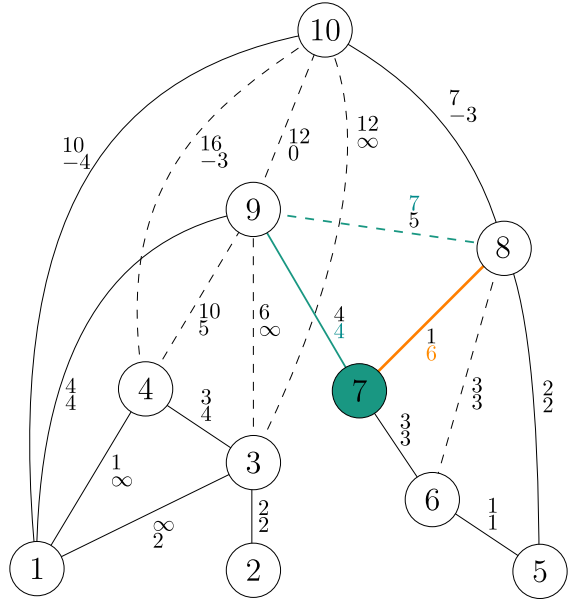
Customize for vertex 7



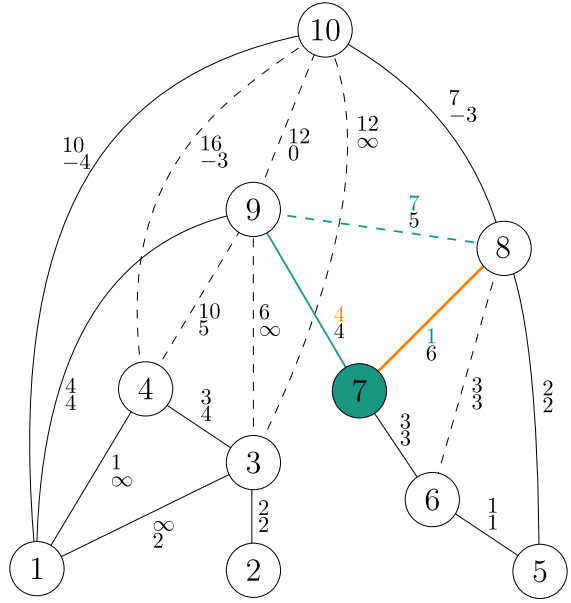
$\{7, 8, 9\}$ is a upper triangle of $(7, 8)$ and a middle triangle of $(7, 9)$



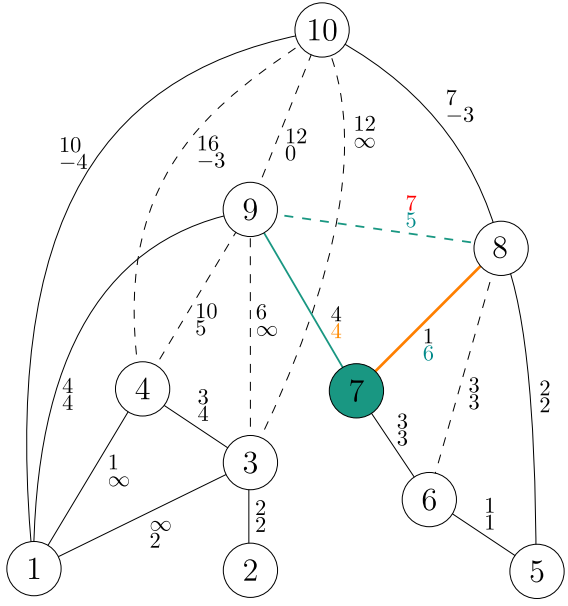
Update upweight for upper triangle if possible



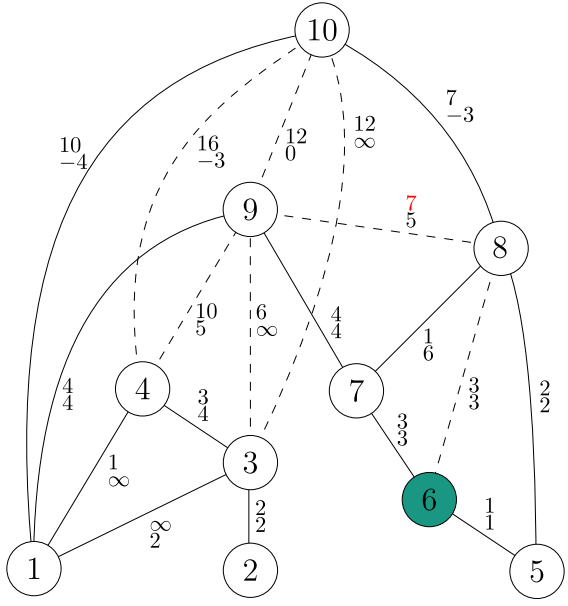
Update downweight for upper triangle if possible



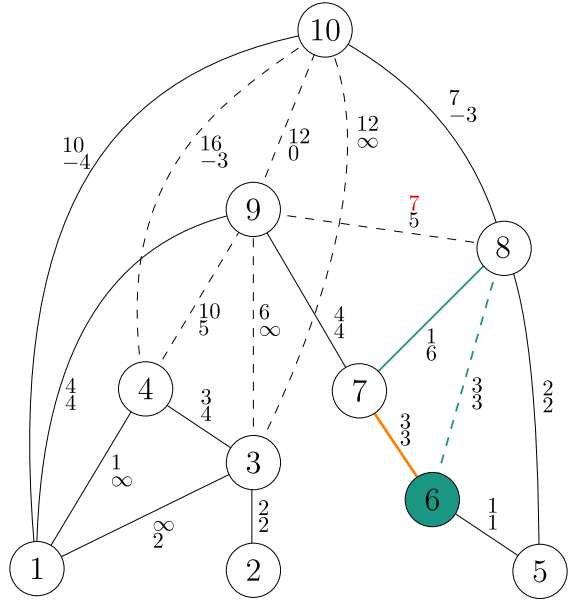
Update upweight for middle triangle if possible



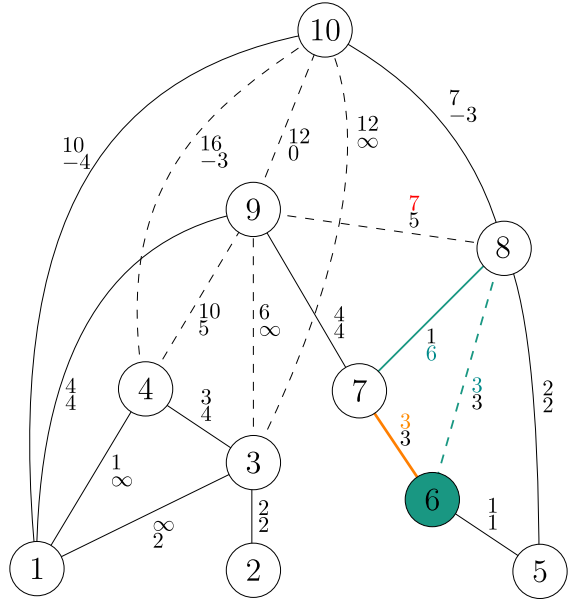
Update downweight for middle triangle if possible



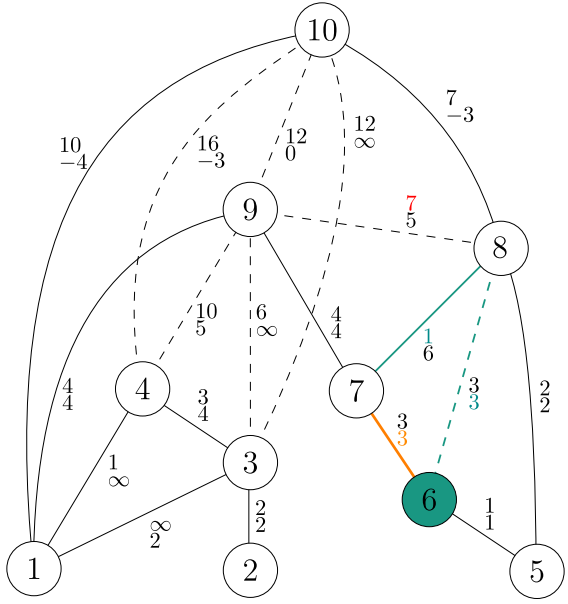
Customize for vertex 6



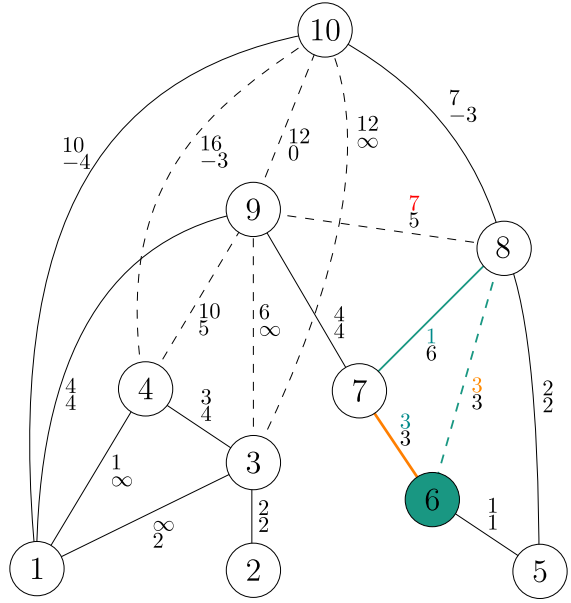
$\{6, 7, 8\}$ is a upper triangle of $(6, 7)$ and a middle triangle of $(6, 8)$



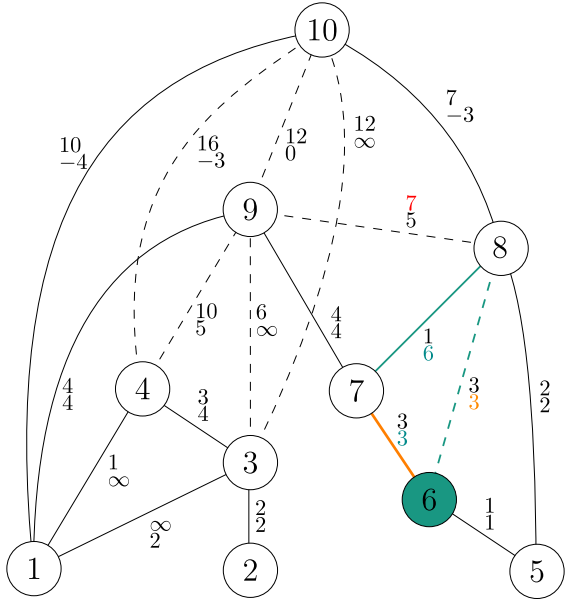
Update upweight for upper triangle if possible



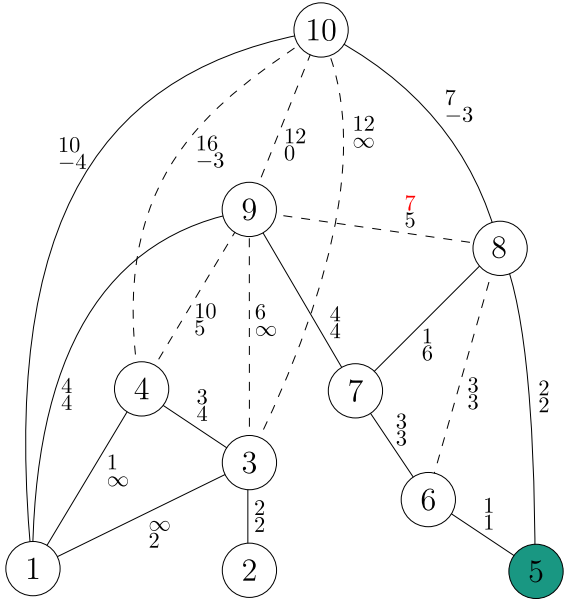
Update downweight for upper triangle if possible



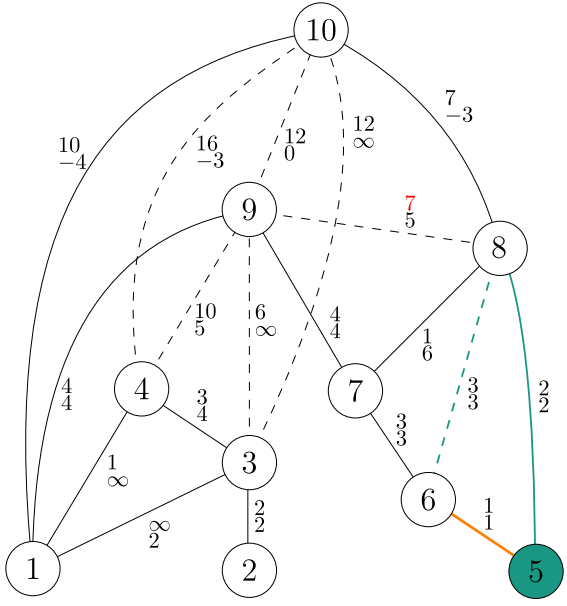
Update upweight for middle triangle if possible



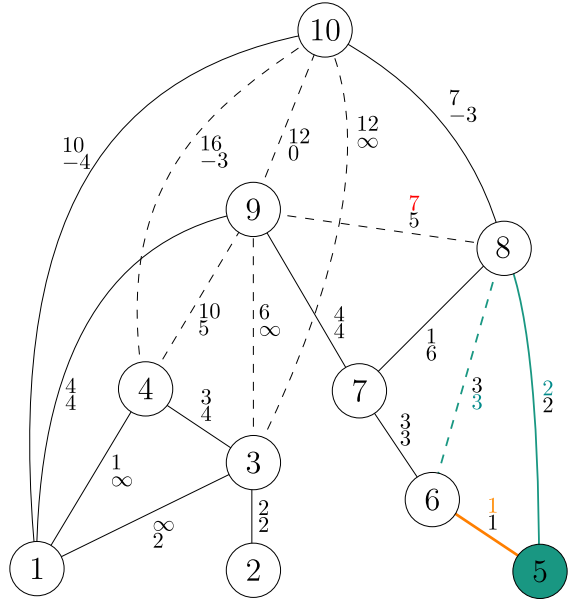
Update downweight for middle triangle if possible



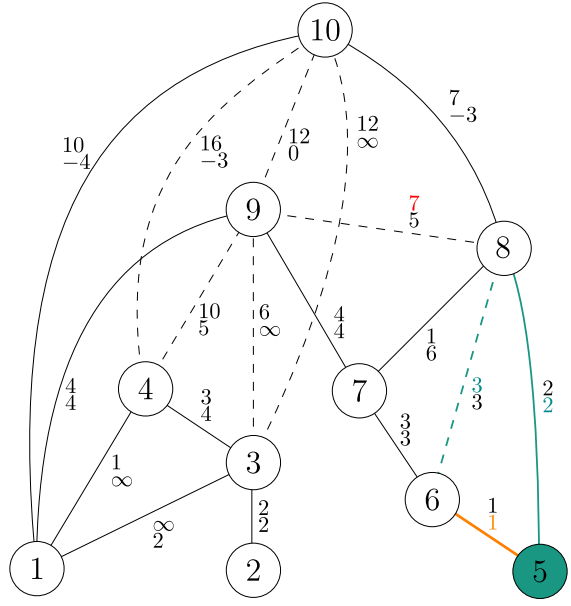
Customize for vertex 5



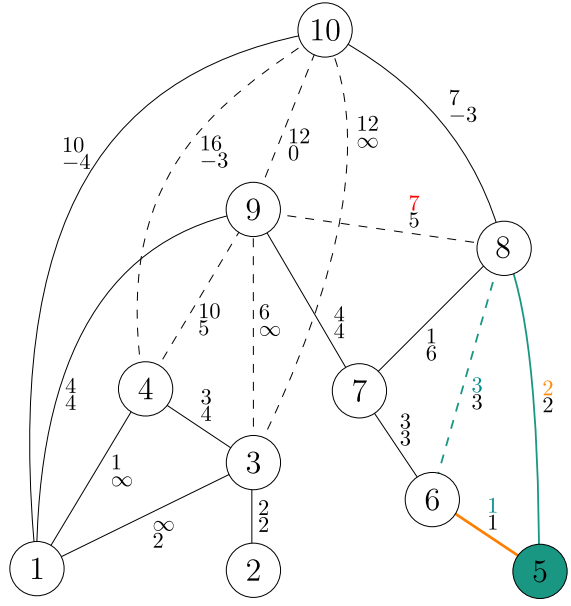
$\{5, 6, 8\}$ is a upper triangle of $(5, 6)$ and a middle triangle of $(5, 8)$



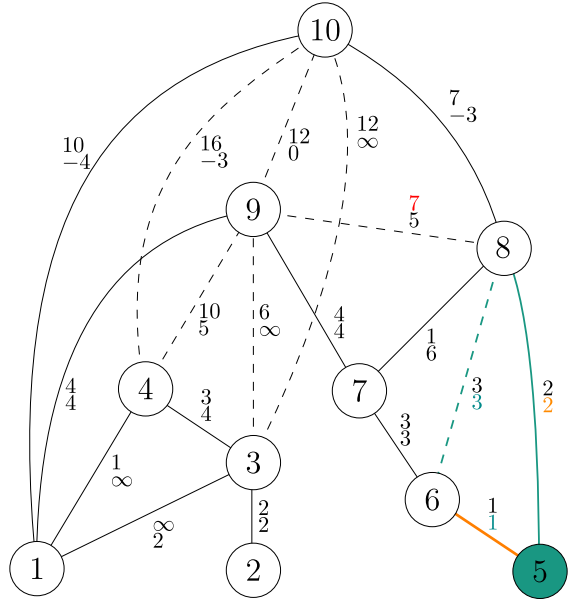
Update upweight for upper triangle if possible



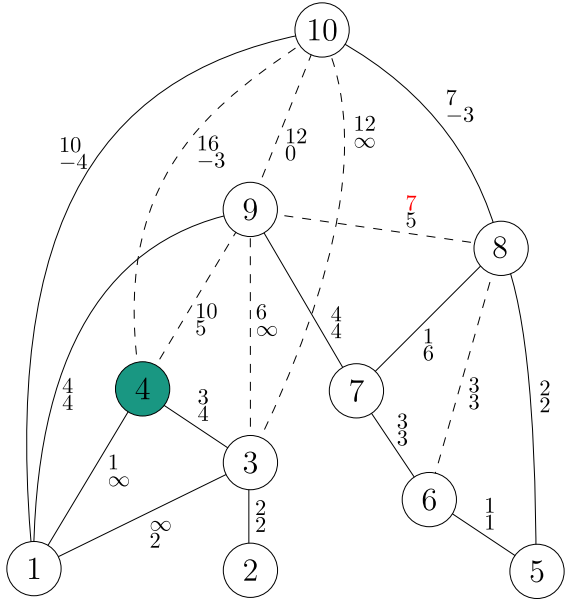
Update downweight for upper triangle if possible



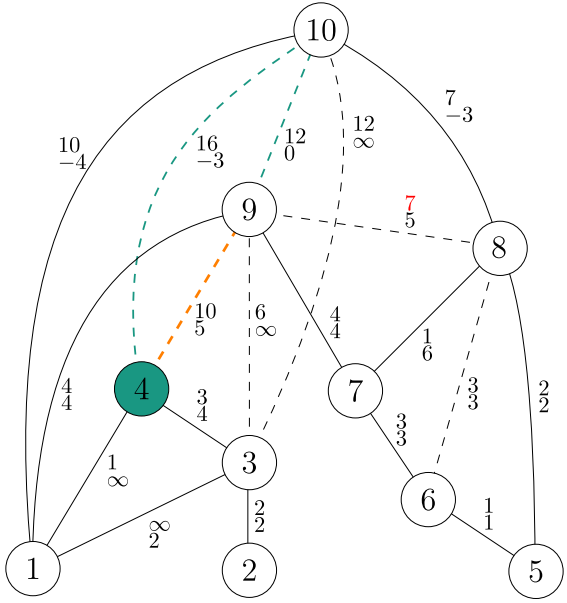
Update upweight for middle triangle if possible



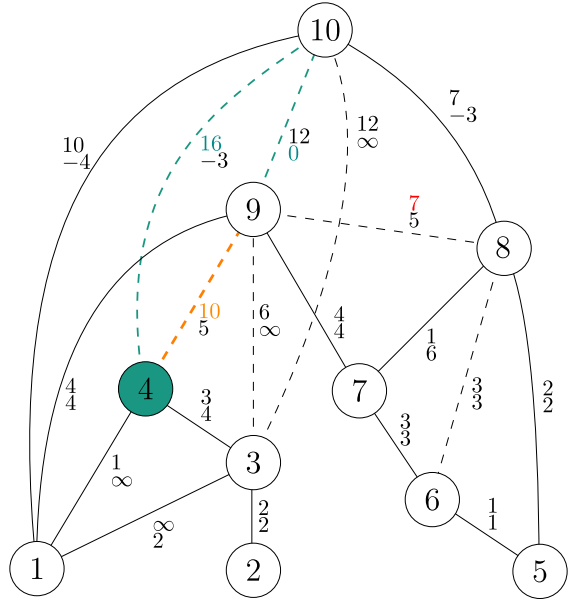
Update downweight for middle triangle if possible



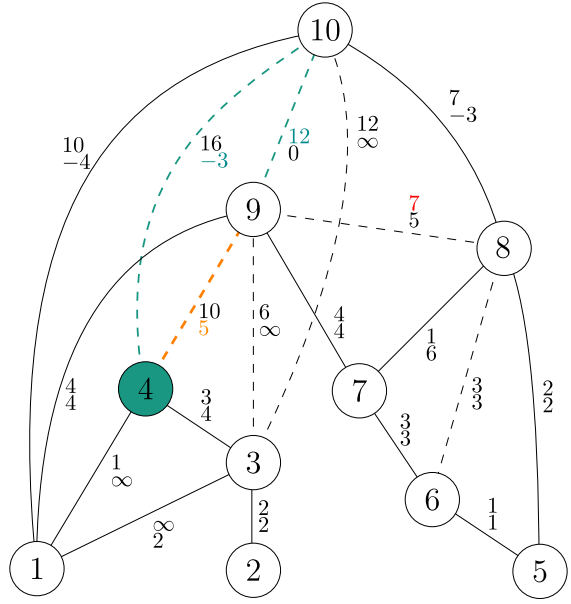
Customize for vertex 4



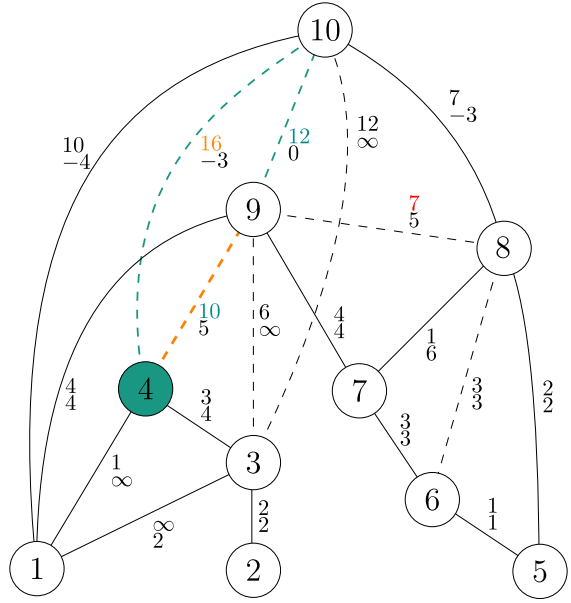
$\{4, 9, 10\}$ is a upper triangle of $(4, 9)$ and a middle triangle of $(4, 10)$



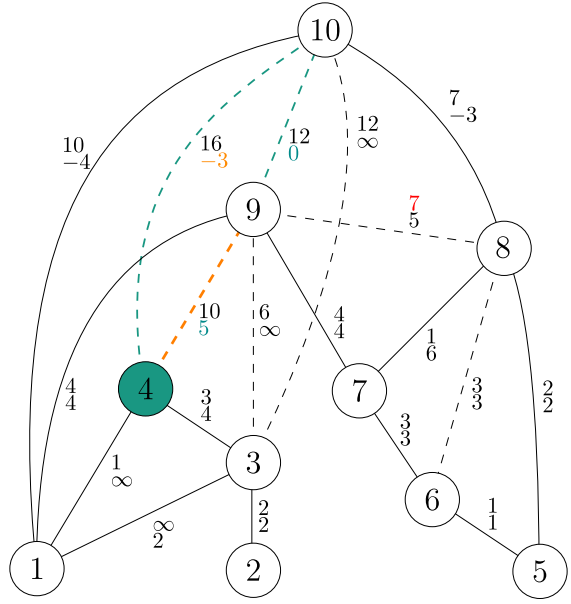
Update upweight for upper triangle if possible



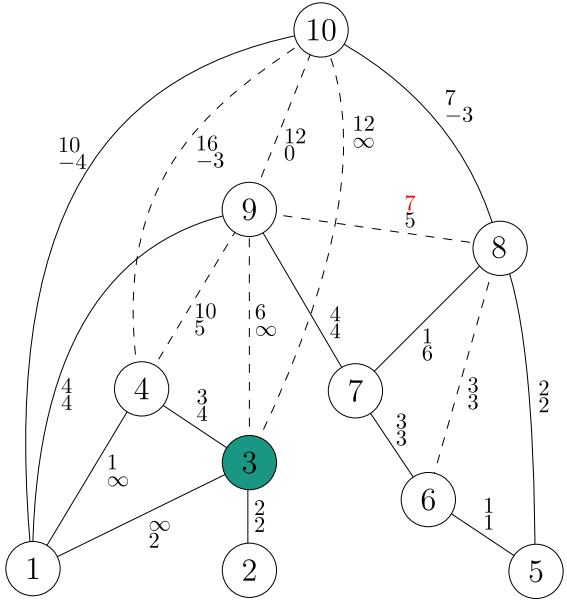
Update downweight for upper triangle if possible



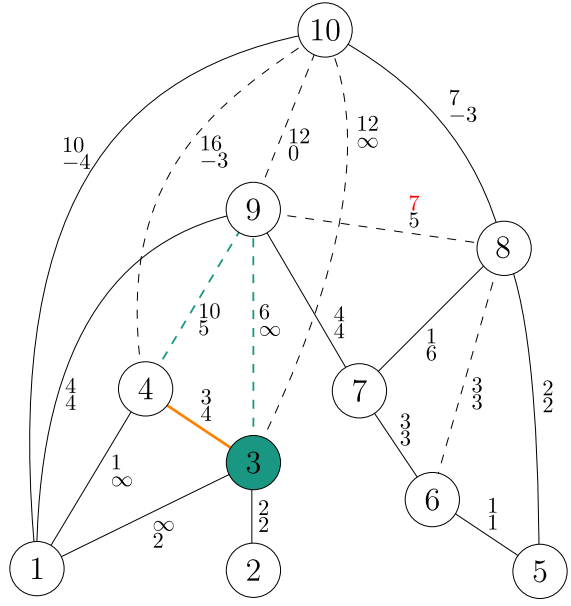
Update upweight for middle triangle if possible



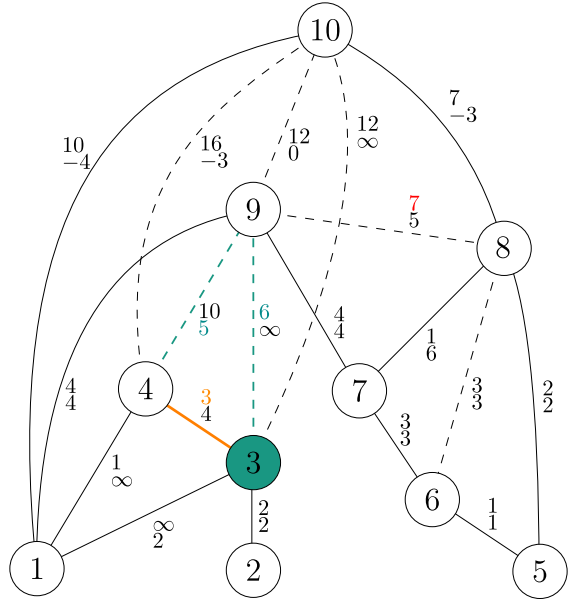
Update downweight for middle triangle if possible



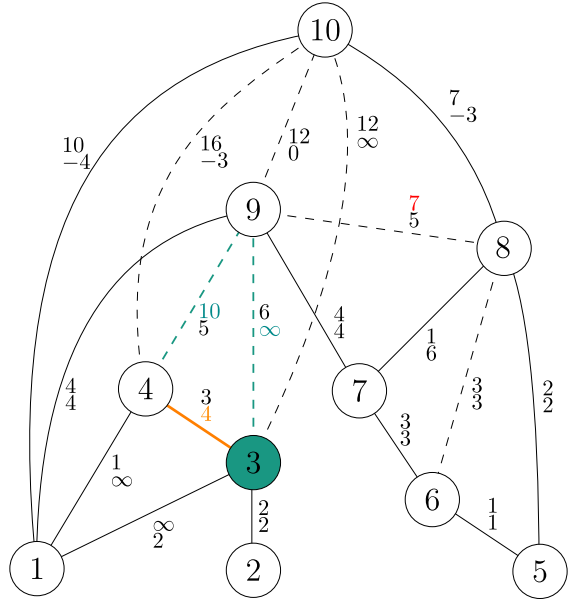
Customize for vertex 3



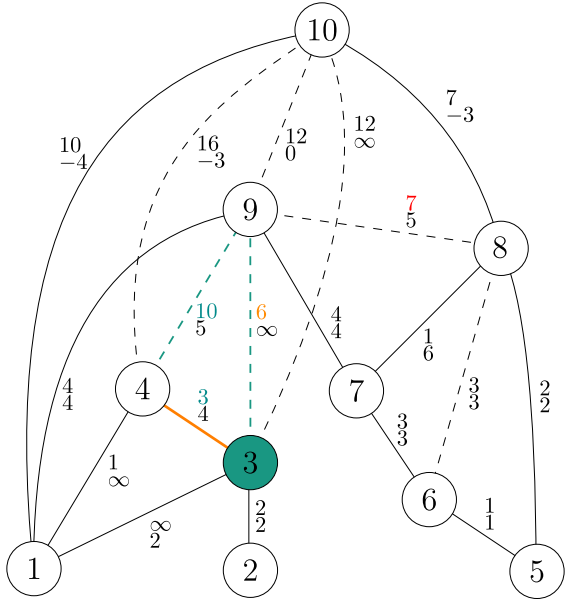
$\{3, 4, 9\}$ is a upper triangle of $(3, 4)$ and a middle triangle of $(3, 9)$



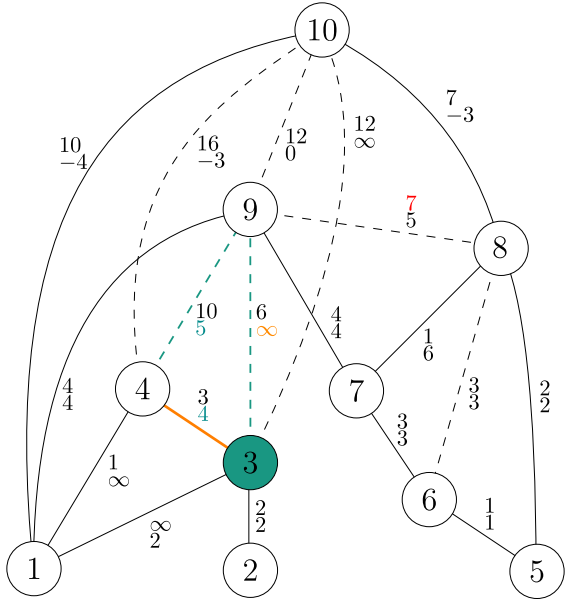
Update upweight for upper triangle if possible



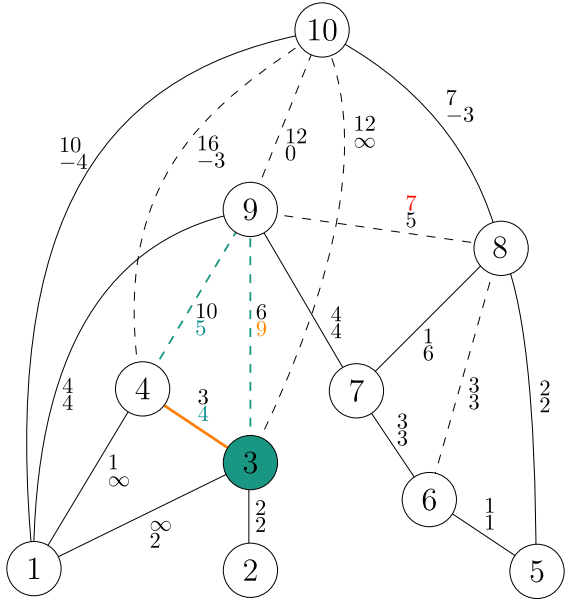
Update downweight for upper triangle if possible

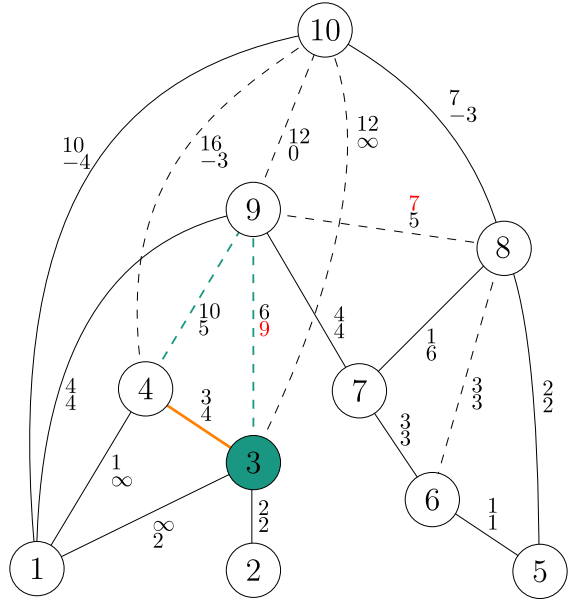


Update upweight for middle triangle if possible

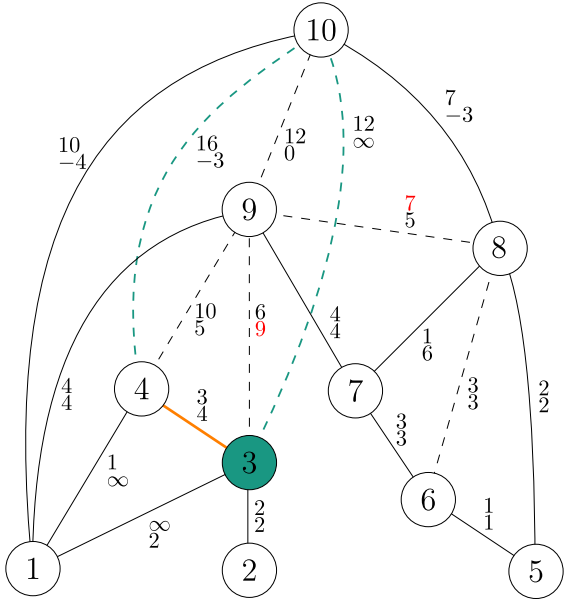


Update downweight for middle triangle if possible

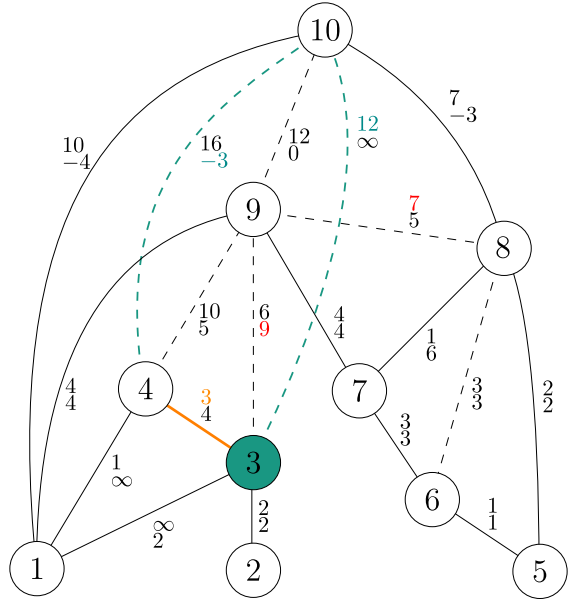

10
9
8
4
7
3
6
1
2
5
10
−4
16
−3
12
0
12
∞
7
−3
7
5
10
5
6
9
4
4
1
6
4
4
3
4
3
3
3
3
2
2
1
∞
1
∞
2
2
2
1
1



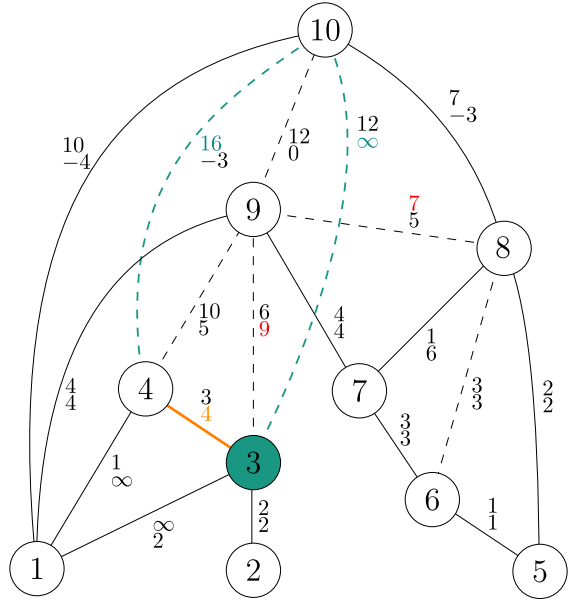
Mark edge as deleted because its weight was changed



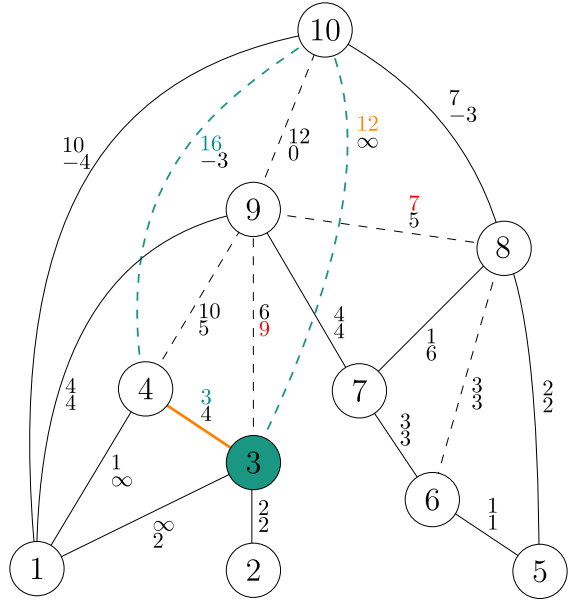
$\{3, 4, 10\}$ is a upper triangle of $(3, 4)$ and a middle triangle of $(3, 10)$



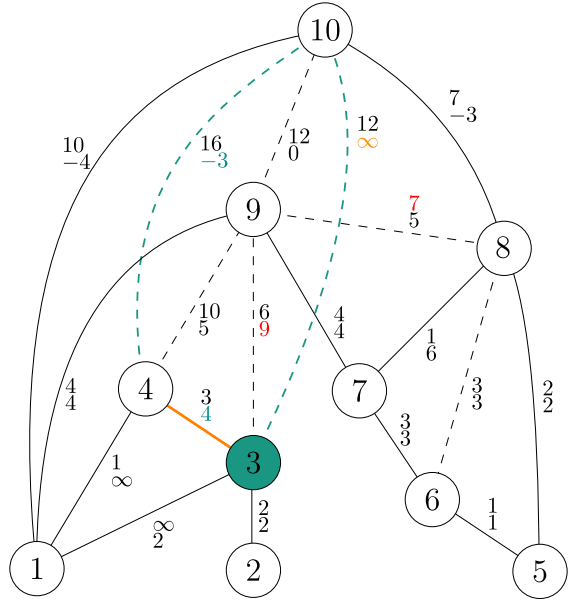
Update upweight for upper triangle if possible



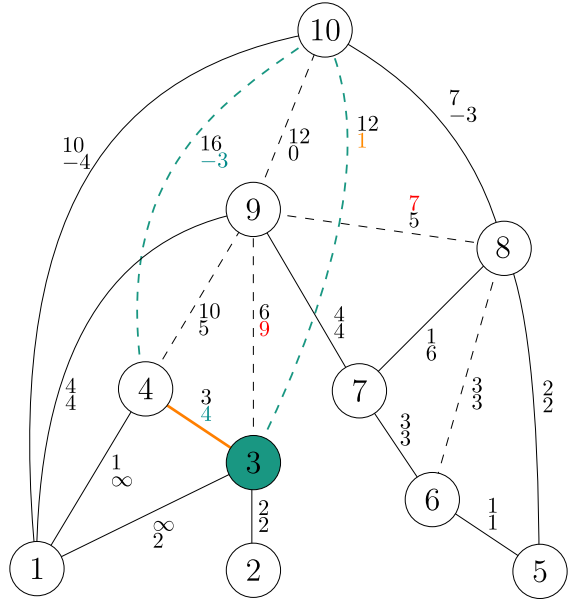
Update downweight for upper triangle if possible

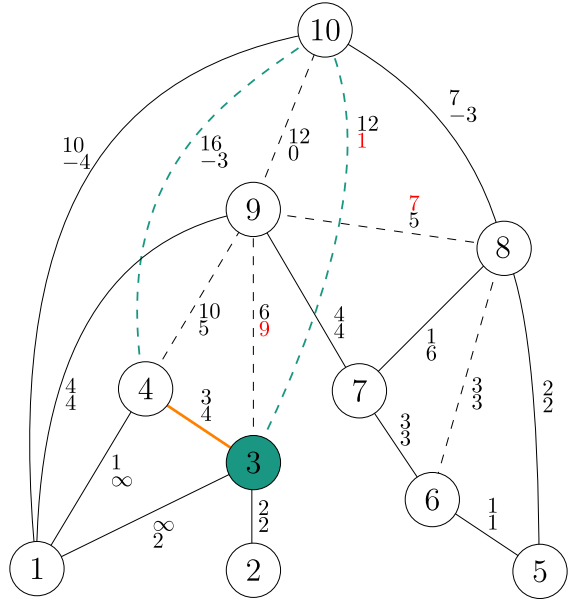


Update upweight for middle triangle if possible

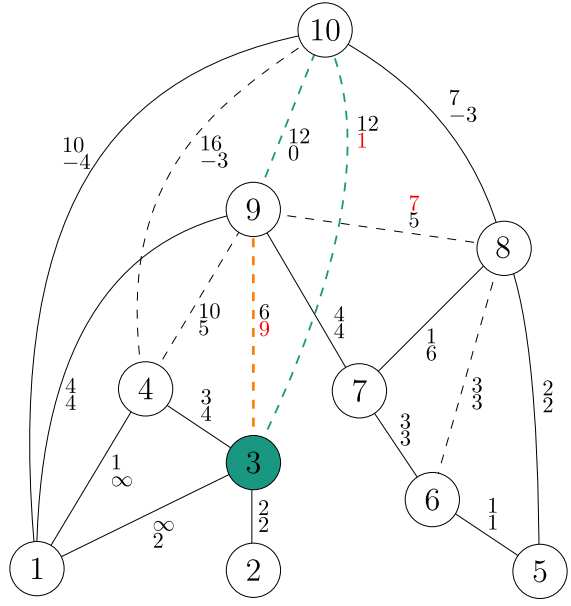


Update downweight for middle triangle if possible

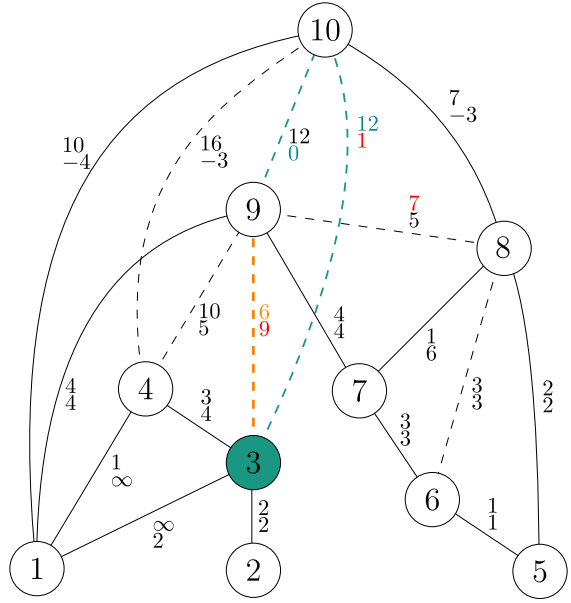

10
9
8
7
4
3
6
5
1
2
10
−4
16
−3
12
0
12
1
7
−3
7
5
10
5
6
9
4
4
1
6
4
4
3
4
3
3
3
3
2
2
1
∞
∞
2
2
2
1
1



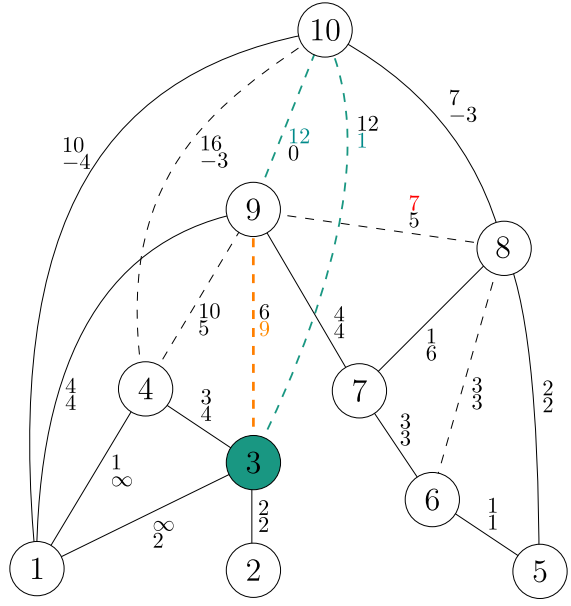
Mark edge as deleted because its weight was changed



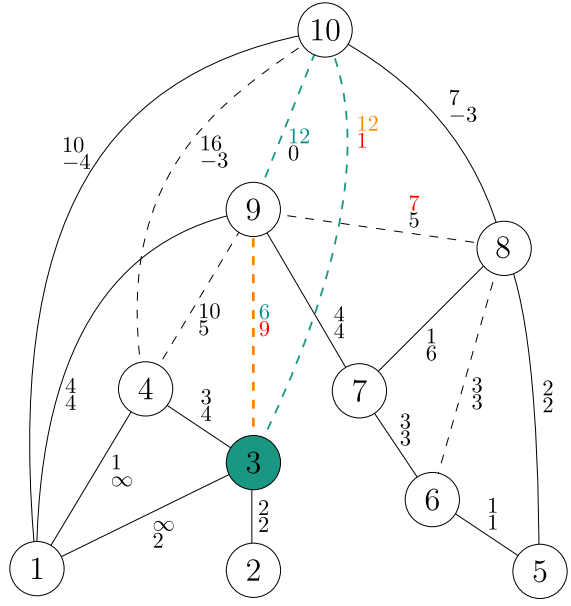
$\{3, 9, 10\}$ is a upper triangle of $(3, 9)$ and a middle triangle of $(3, 10)$



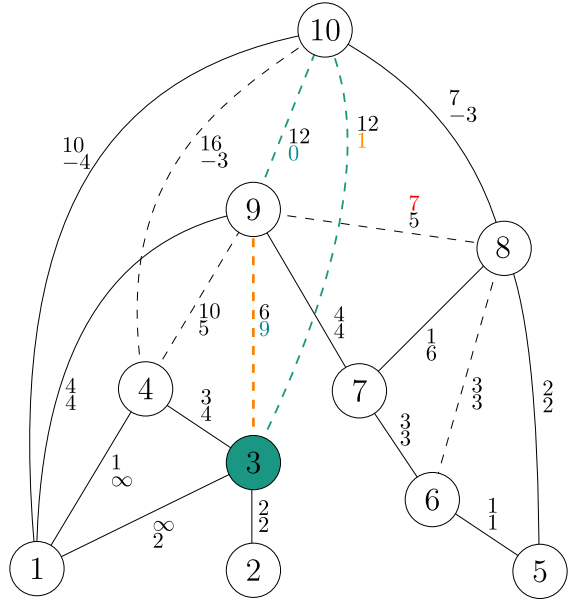
Update upweight for upper triangle if possible



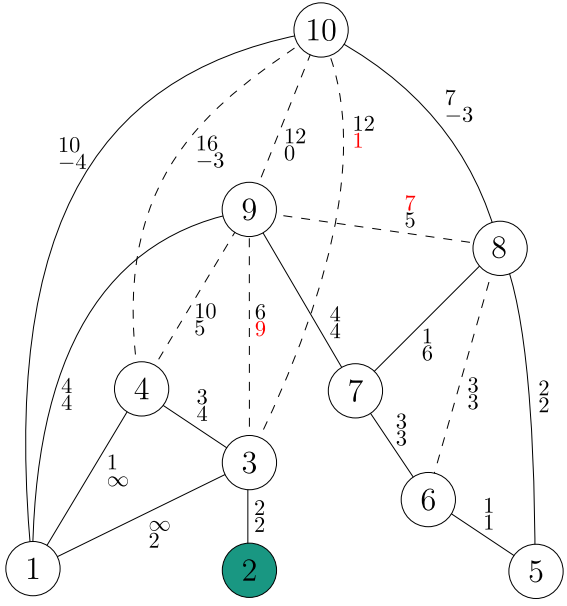
Update downweight for upper triangle if possible



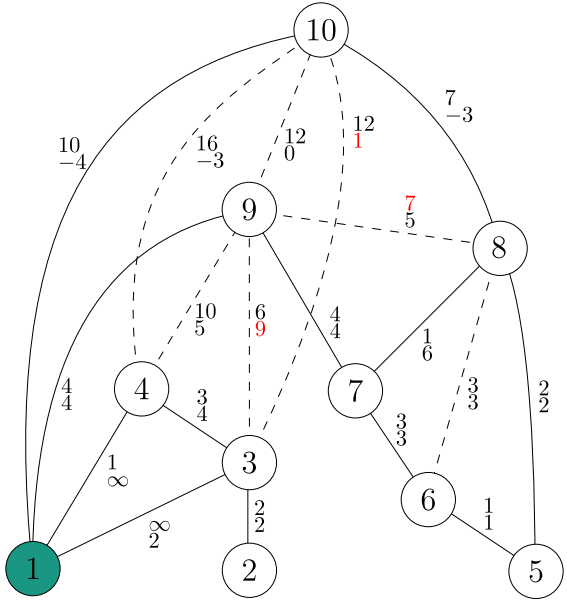
Update upweight for middle triangle if possible



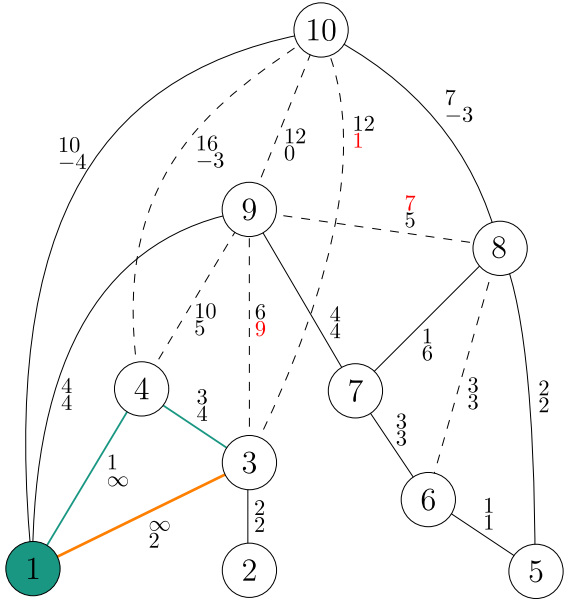
Update downweight for middle triangle if possible



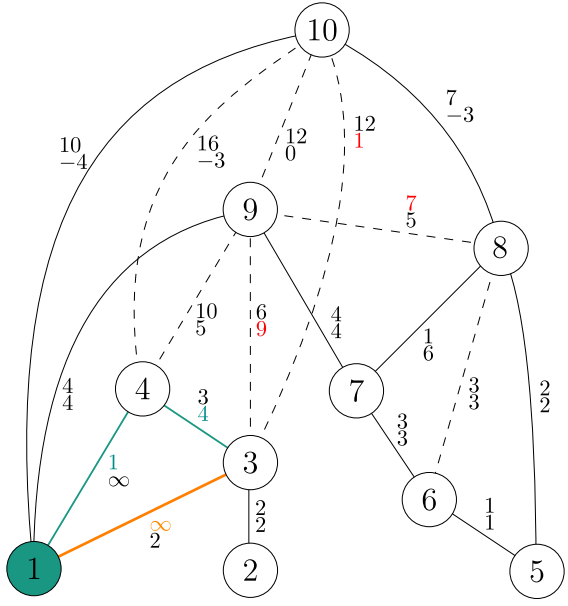
Customize for vertex 2



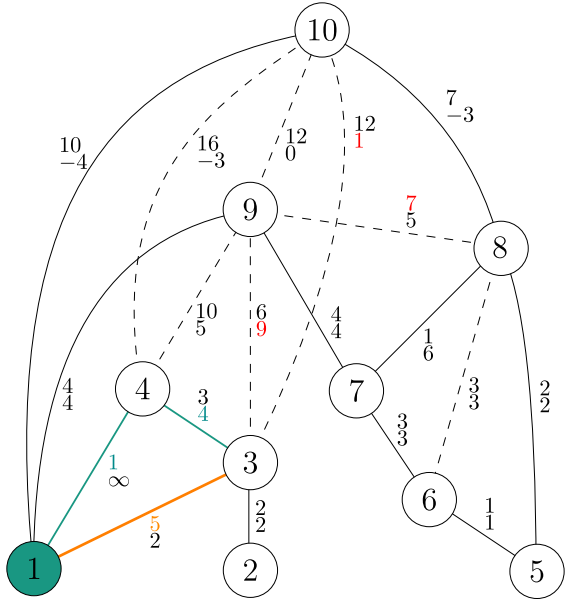
Customize for vertex 1

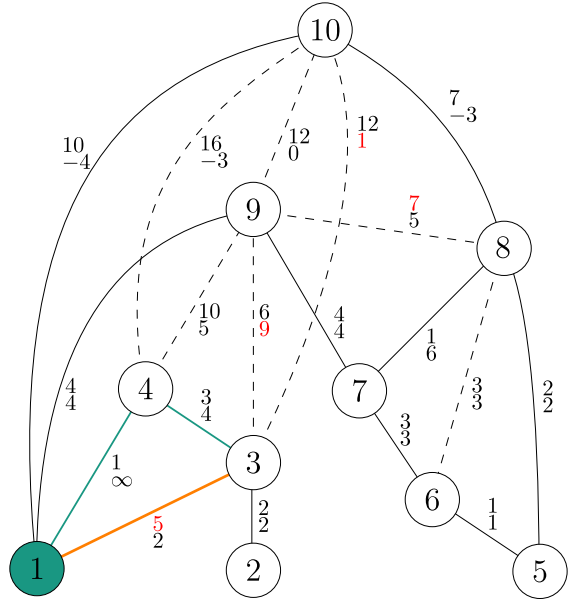


$\{1, 3, 4\}$ is a upper triangle of $(1, 3)$ and a middle triangle of $(1, 4)$

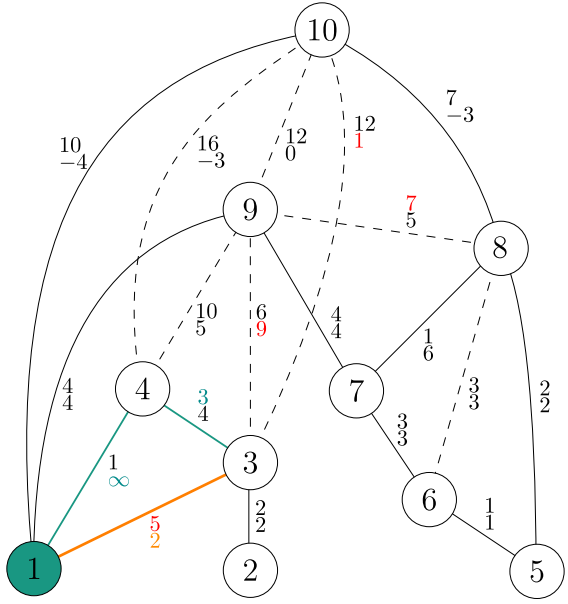


Update upweight for upper triangle if possible

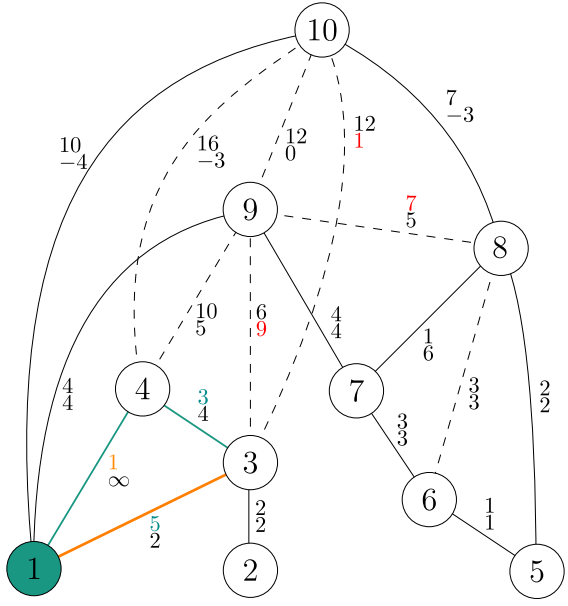

10
9
8
4
7
3
6
5
2
1
10
−4
16
−3
12
0
12
1
7
−3
7
5
10
5
6
9
4
4
1
6
4
4
3
4
3
3
3
3
2
2
1
∞
5
2
2
2
1
1



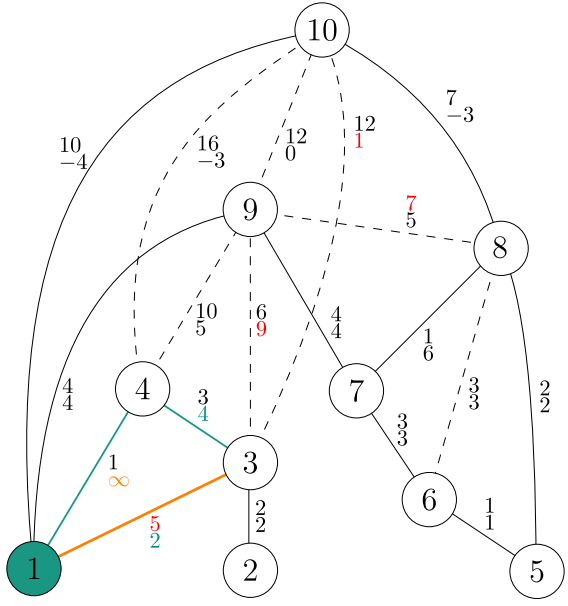
Mark edge as deleted because its weight was changed



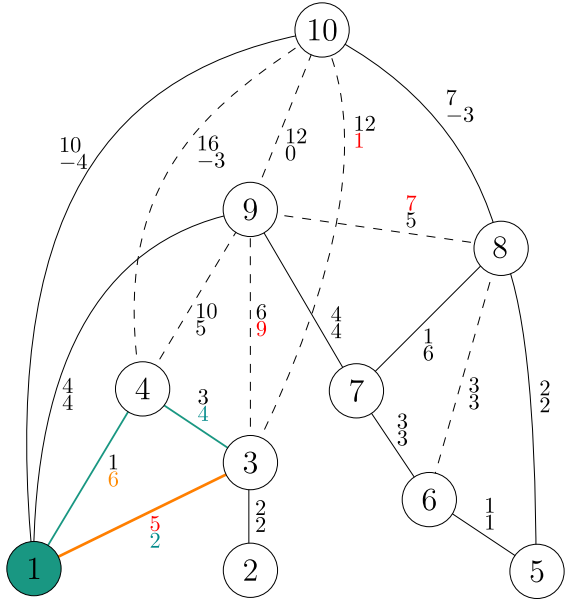
Update downweight for upper triangle if possible

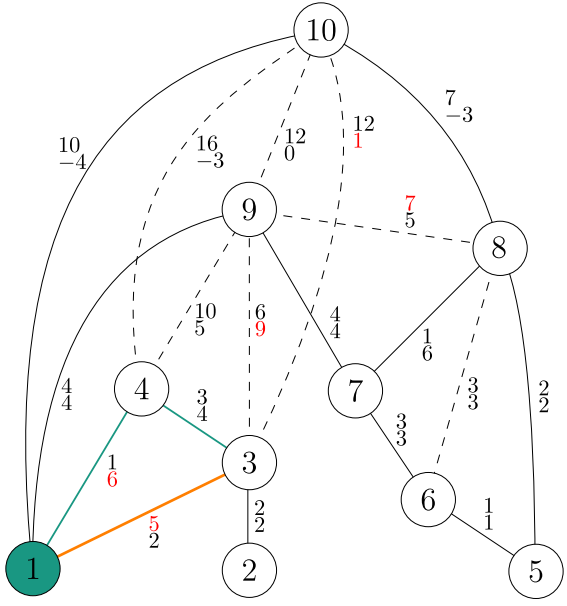


Update upweight for middle triangle if possible

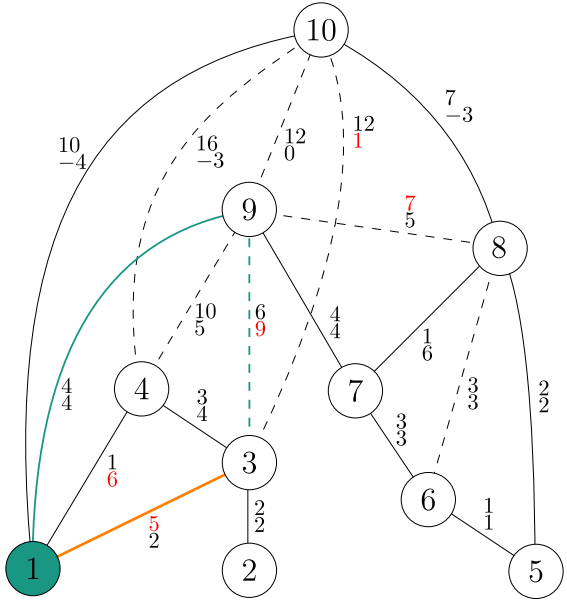


Update downweight for middle triangle if possible

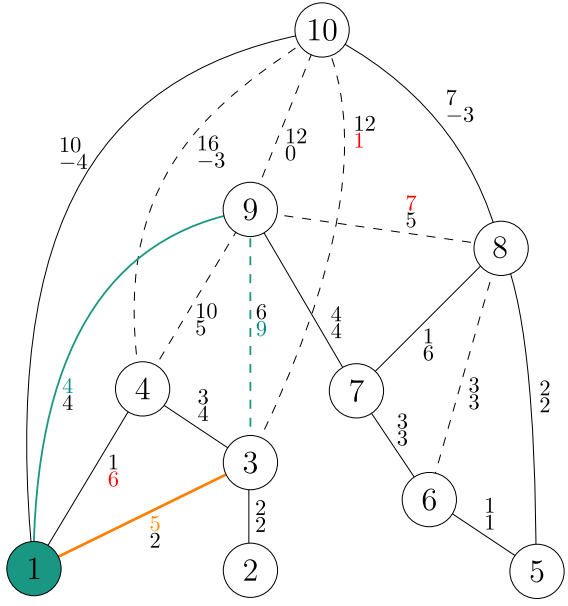

10
9
8
4
7
3
6
5
1
2
10
−4
16
−3
12
0
12
1
7
−3
7
5
10
5
6
9
4
4
1
6
4
4
3
4
3
3
3
3
2
2
1
6
5
2
2
2
1
1



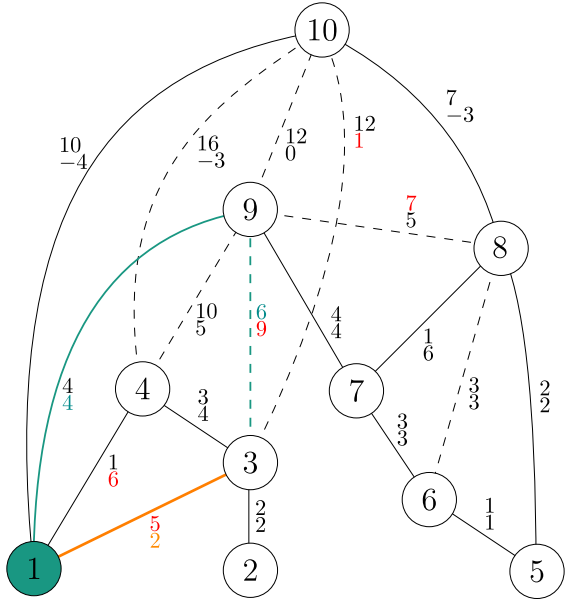
Mark edge as deleted because its weight was changed



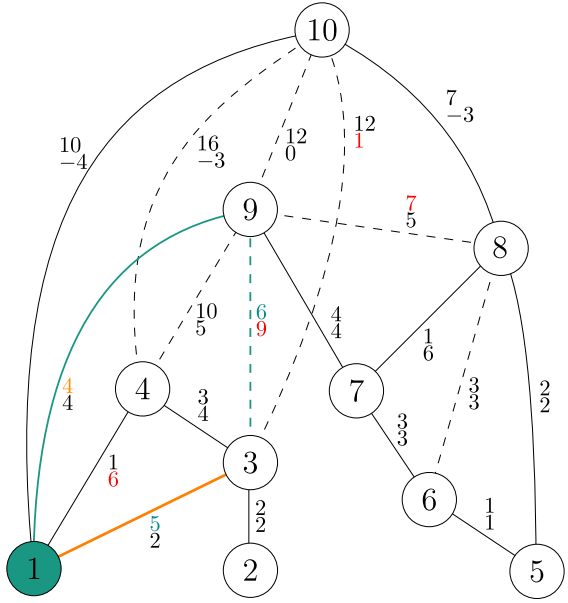
$\{1, 3, 9\}$ is a upper triangle of $(1, 3)$ and a middle triangle of $(1, 9)$



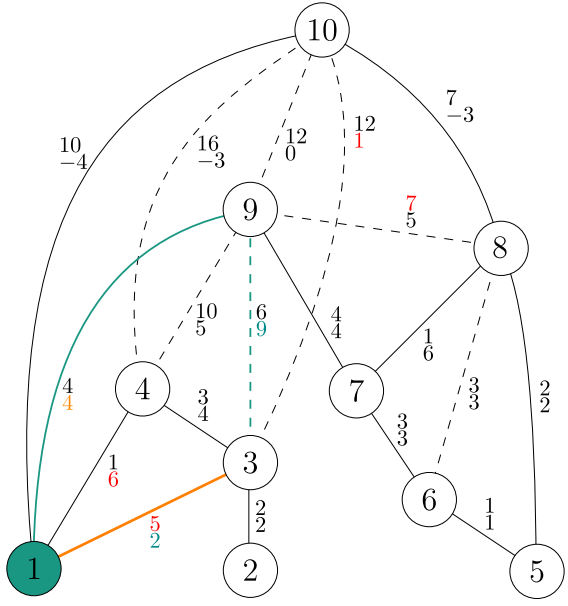
Update upweight for upper triangle if possible



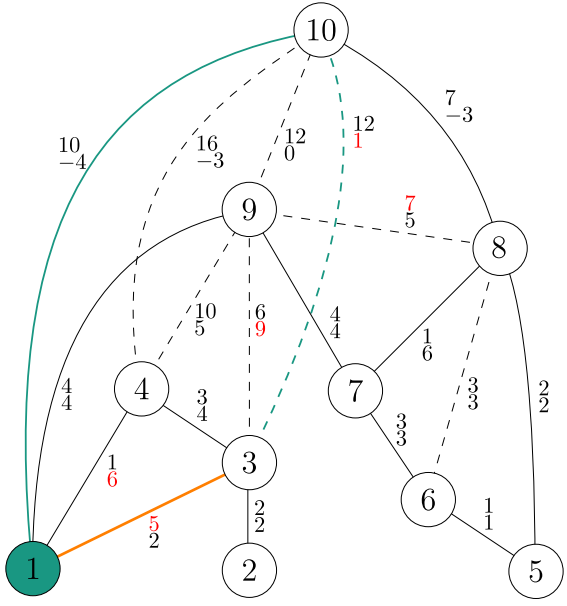
Update downweight for upper triangle if possible



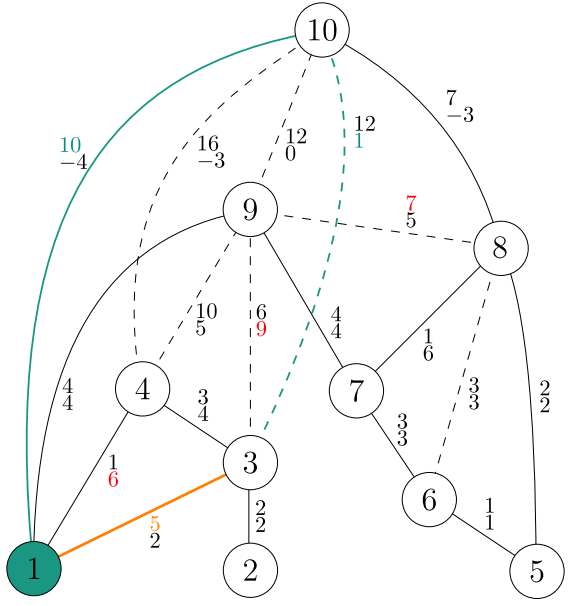
Update upweight for middle triangle if possible



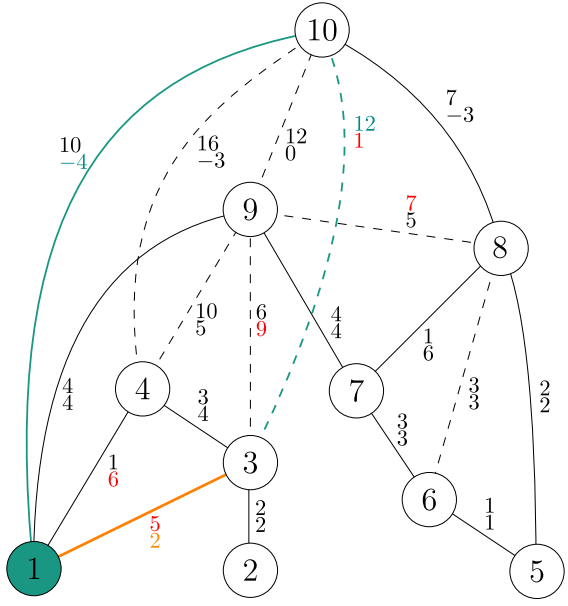
Update downweight for middle triangle if possible



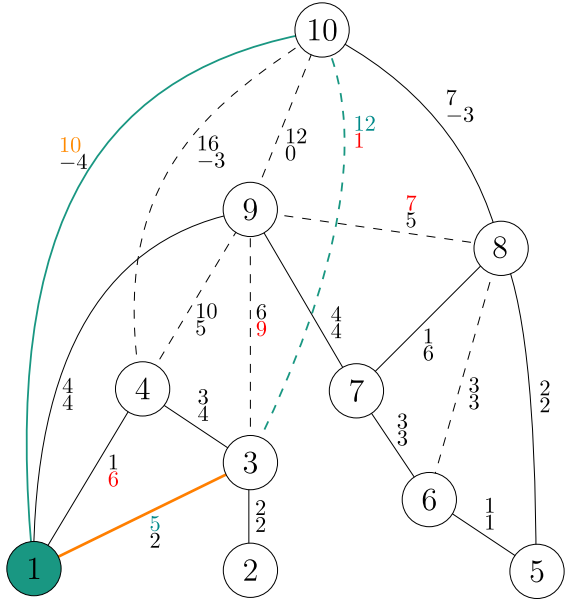
$\{1, 3, 10\}$ is a upper triangle of $(1, 3)$ and a middle triangle of $(1, 10)$



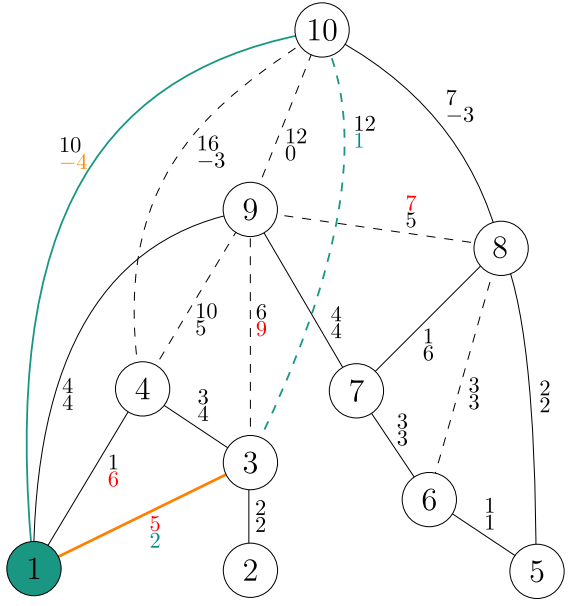
Update upweight for upper triangle if possible



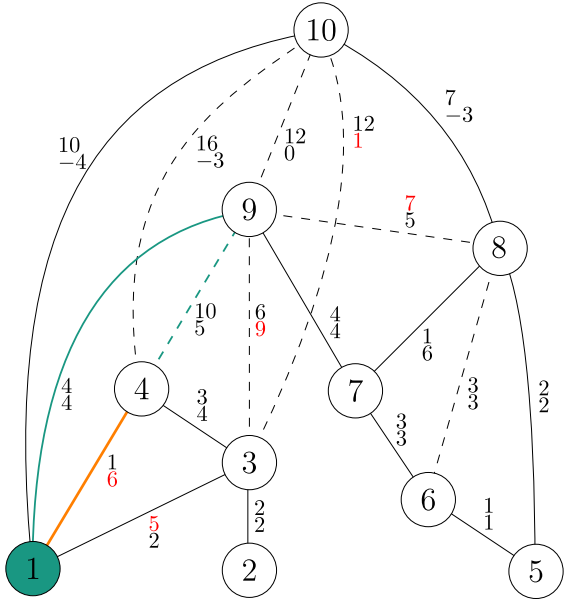
Update downweight for upper triangle if possible



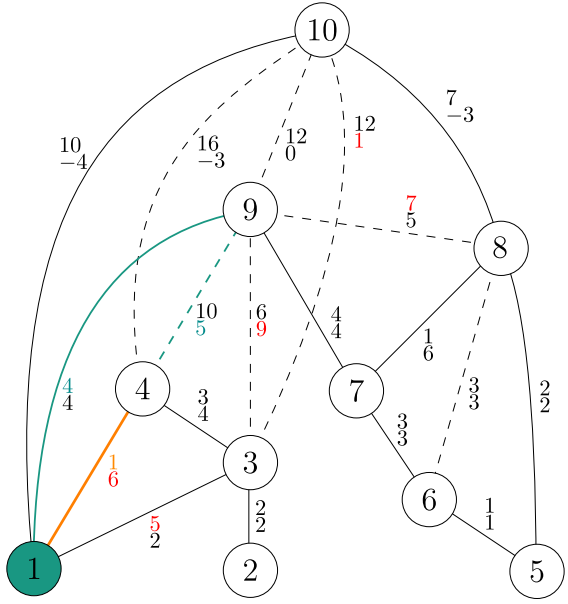
Update upweight for middle triangle if possible



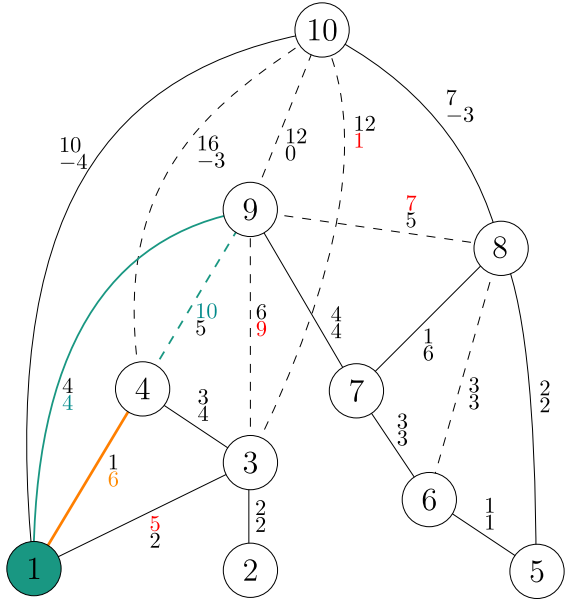
Update downweight for middle triangle if possible



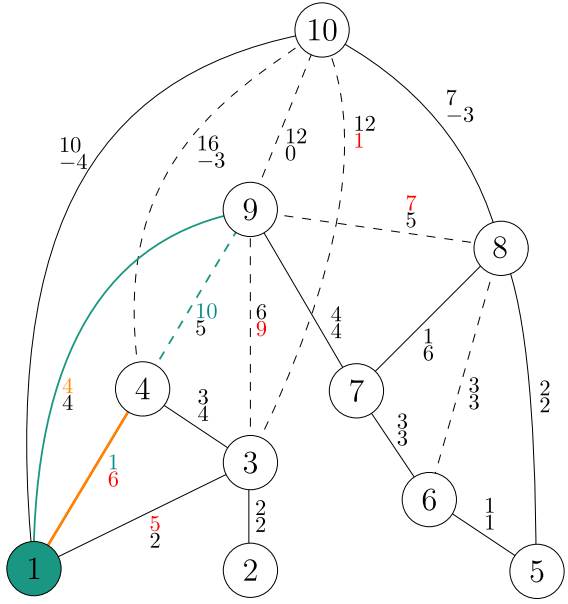
$\{1, 4, 9\}$ is a upper triangle of $(1, 4)$ and a middle triangle of $(1, 9)$



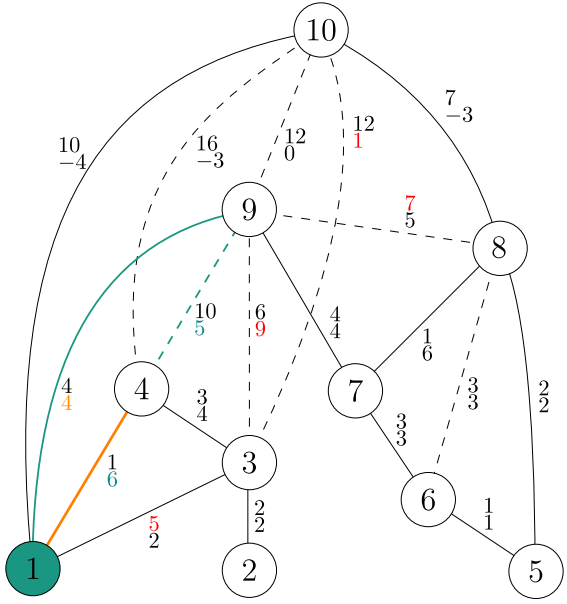
Update upweight for upper triangle if possible



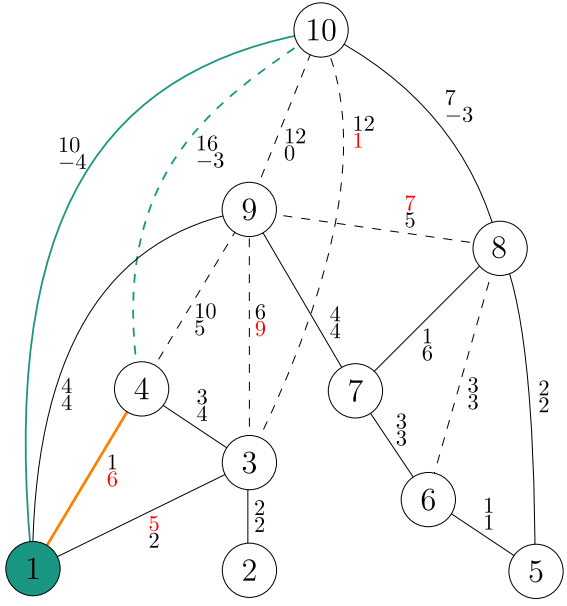
Update downweight for upper triangle if possible



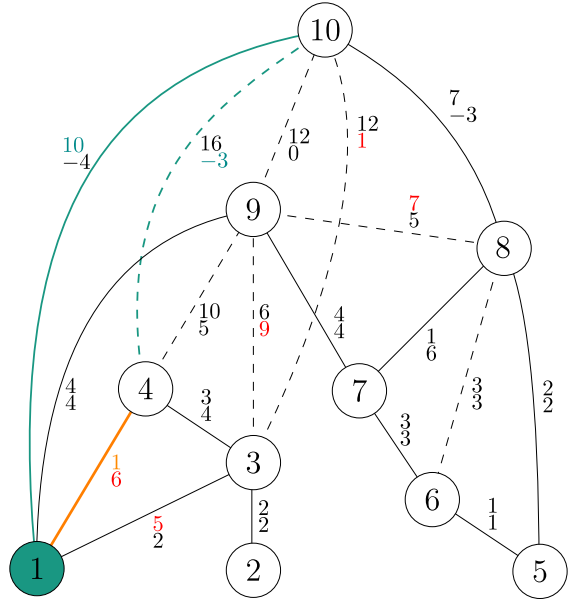
Update upweight for middle triangle if possible



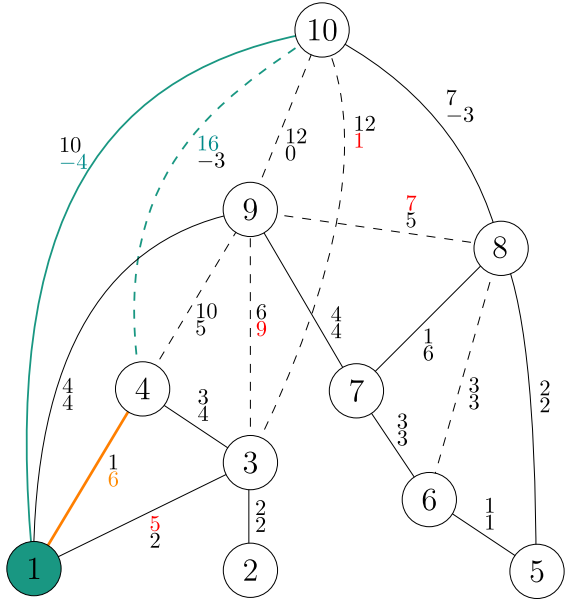
Update downweight for middle triangle if possible



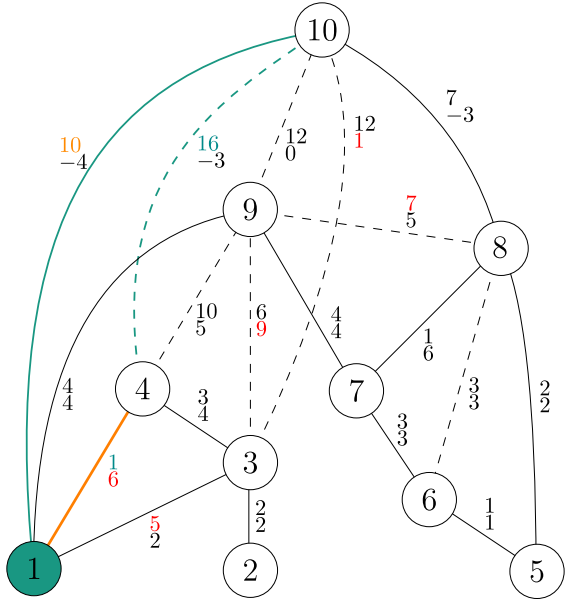
$\{1, 4, 10\}$ is a upper triangle of $(1, 4)$ and a middle triangle of $(1, 10)$



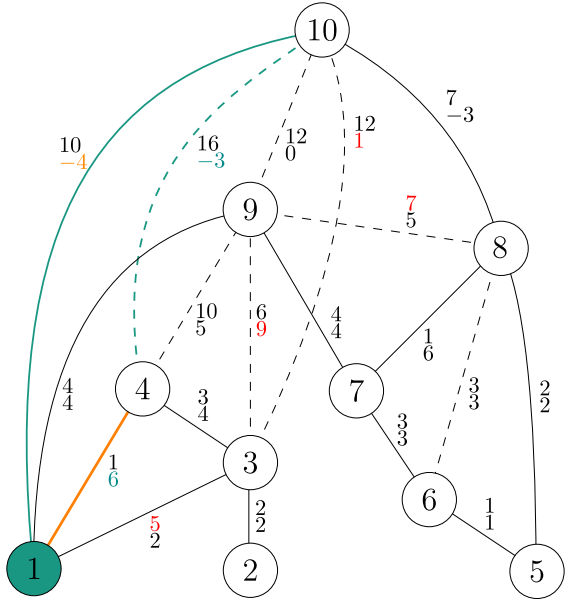
Update upweight for upper triangle if possible



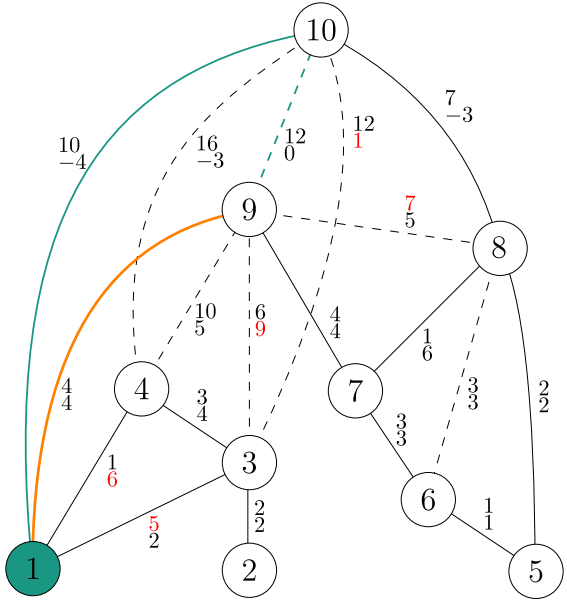
Update downweight for upper triangle if possible



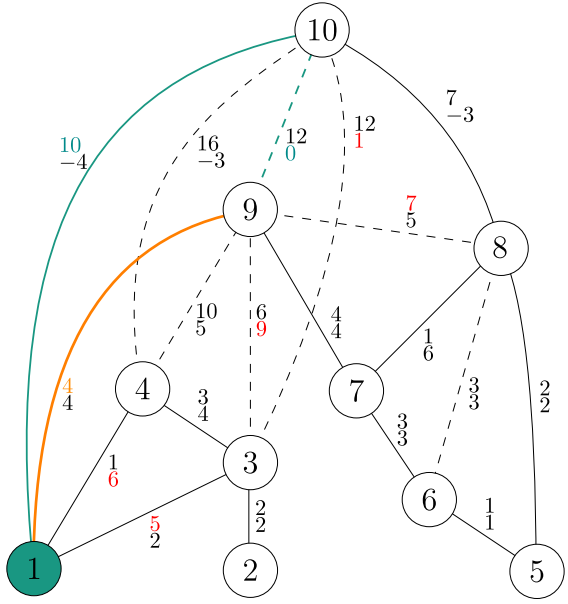
Update upweight for middle triangle if possible



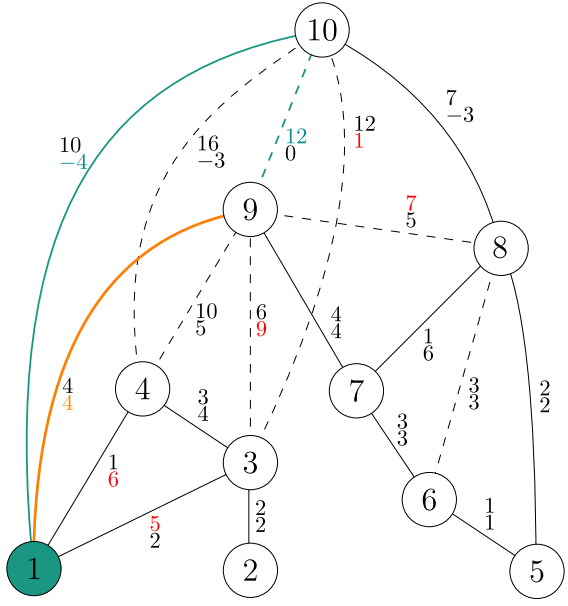
Update downweight for middle triangle if possible



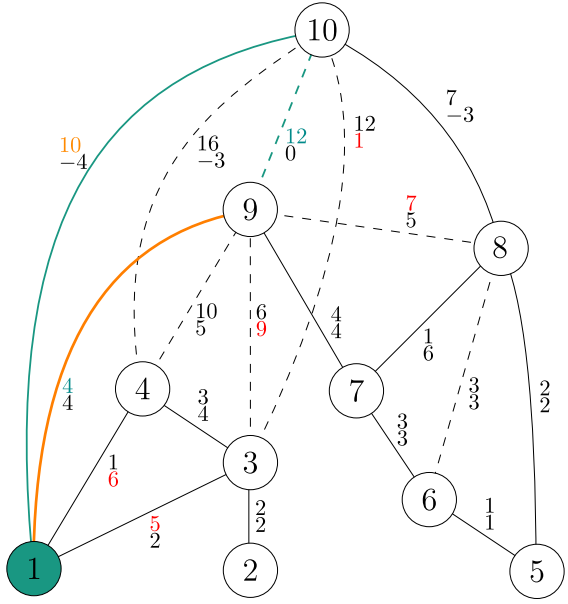
$\{1, 9, 10\}$ is a upper triangle of $(1, 9)$ and a middle triangle of $(1, 10)$



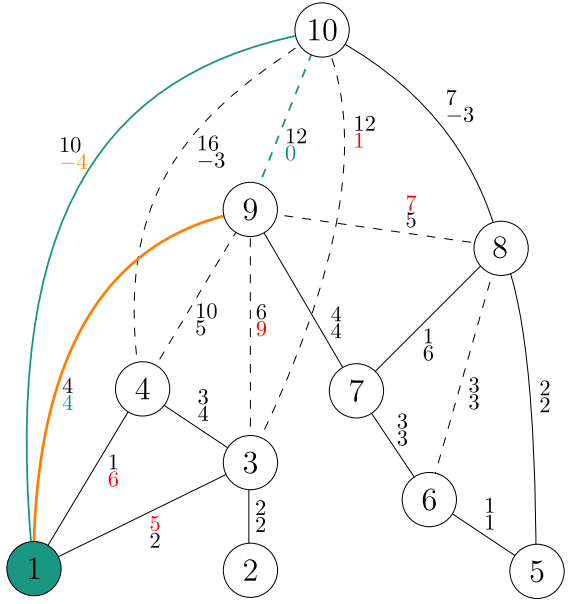
Update upweight for upper triangle if possible



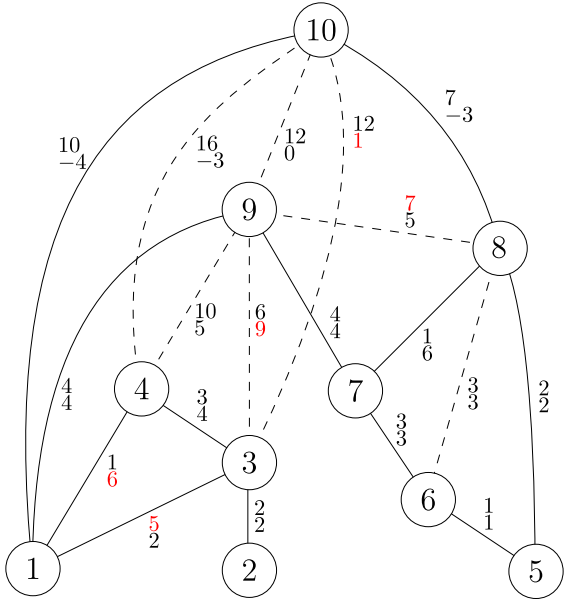
Update downweight for upper triangle if possible



Update upweight for middle triangle if possible

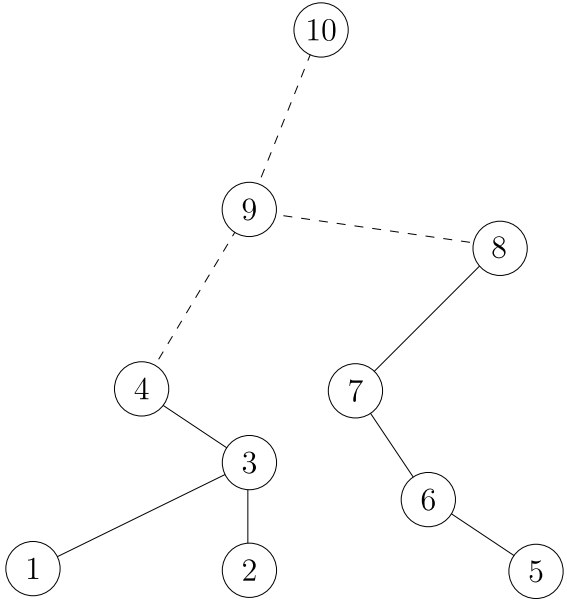


Update downweight for middle triangle if possible

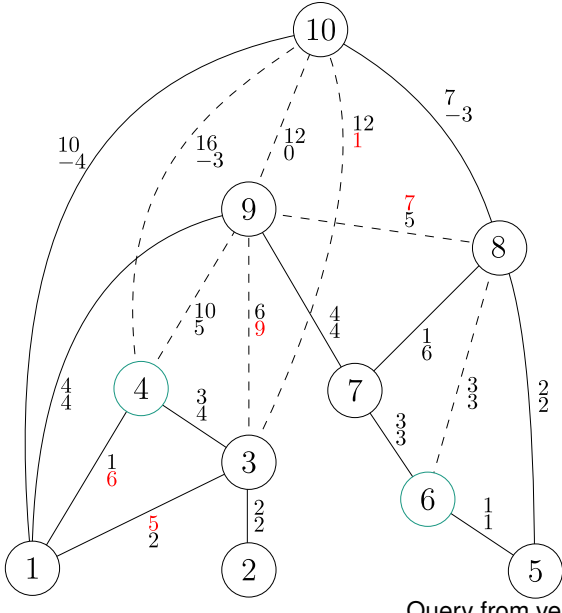


Perfect customization done

Query



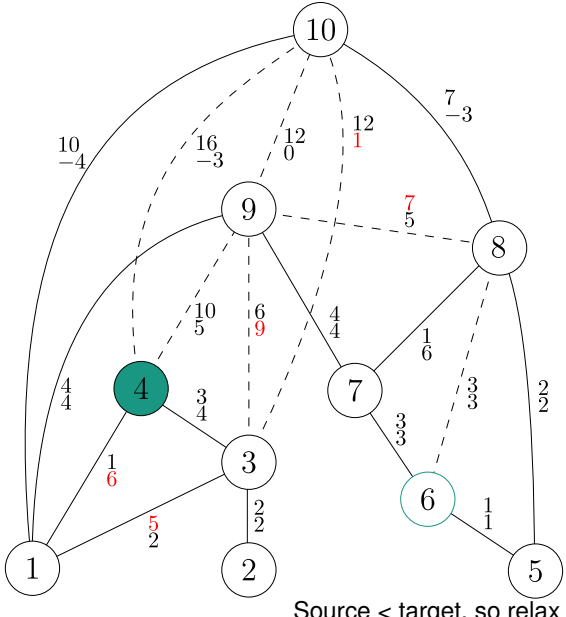
Elimination Tree



Query from vertex 4 to 6

Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	∞	n/a	n/a
8	∞	∞	n/a	n/a
9	∞	∞	n/a	n/a
10	∞	∞	n/a	n/a

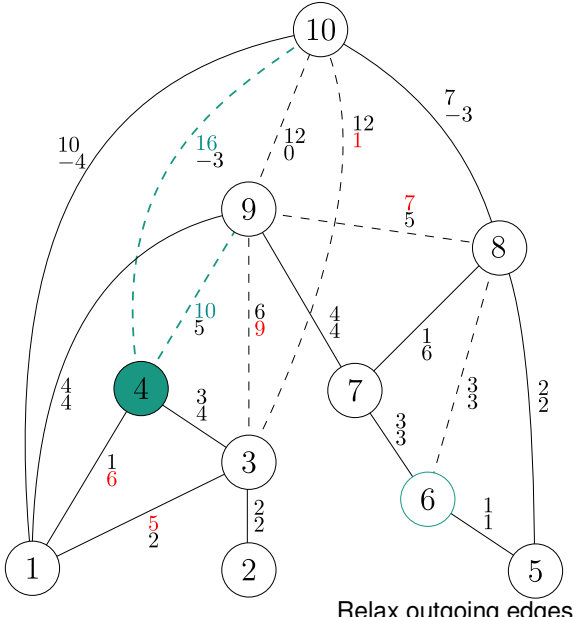
Tentative distance: ∞
Up-down path root: n/a



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	∞	n/a	n/a
8	∞	∞	n/a	n/a
9	∞	∞	n/a	n/a
10	∞	∞	n/a	n/a

Tentative distance: ∞
Up-down path root: n/a

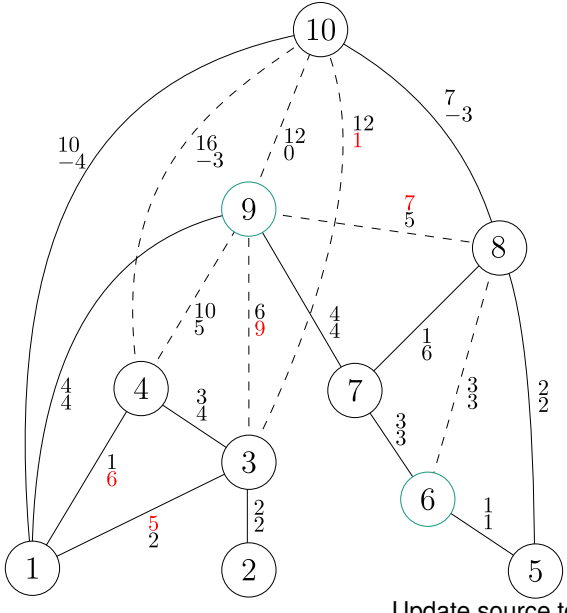
Source < target, so relax forward from source



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	∞	n/a	n/a
8	∞	∞	n/a	n/a
9	10	∞	4	n/a
10	16	∞	4	n/a

Tentative distance: ∞
Up-down path root: n/a

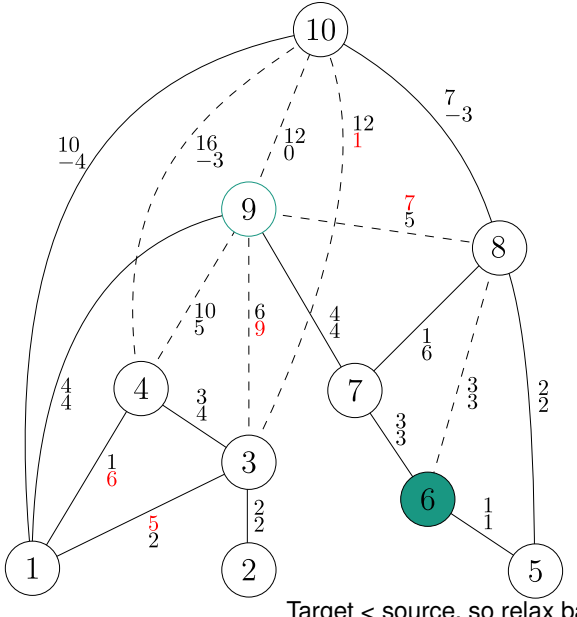
Relax outgoing edges with upweights



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	∞	n/a	n/a
8	∞	∞	n/a	n/a
9	10	∞	4	n/a
10	16	∞	4	n/a

Tentative distance: ∞
Up-down path root: n/a

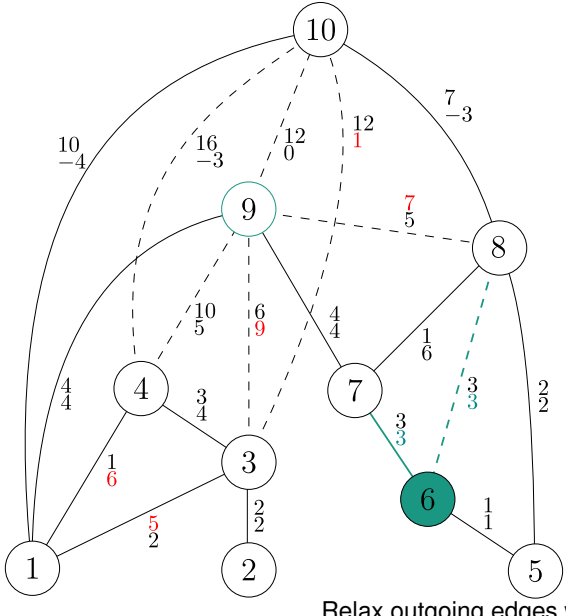
Update source to its parent



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	∞	n/a	n/a
8	∞	∞	n/a	n/a
9	10	∞	4	n/a
10	16	∞	4	n/a

Tentative distance: ∞
Up-down path root: n/a

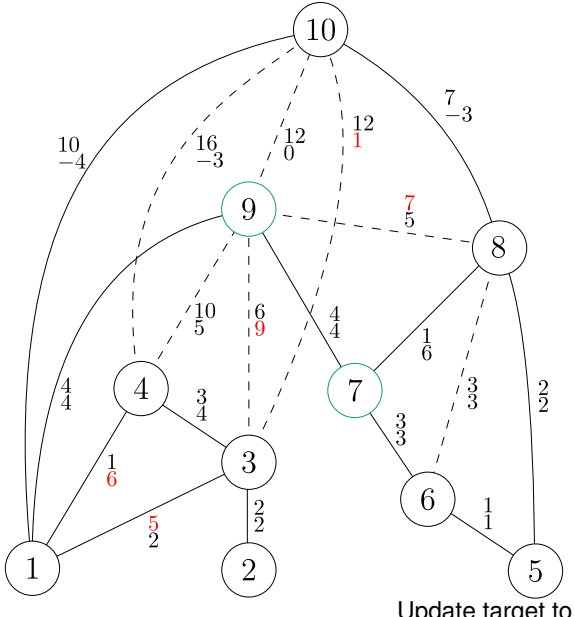
Target < source, so relax backward from target



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	∞	4	n/a
10	16	∞	4	n/a

Tentative distance: ∞
Up-down path root: n/a

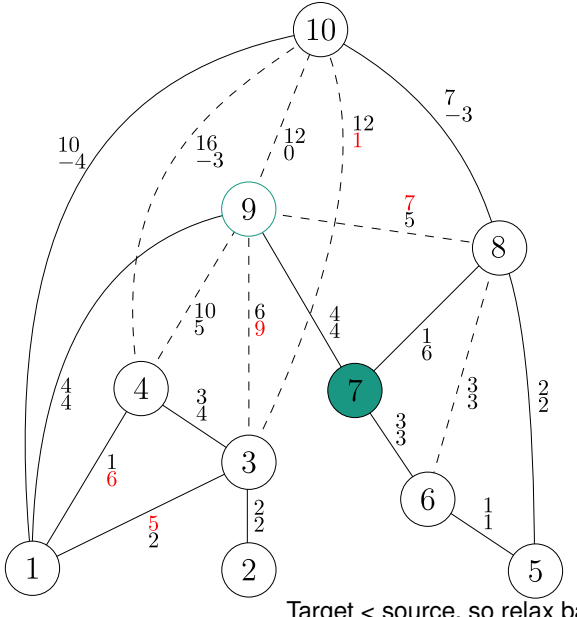
Relax outgoing edges with downweights



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	∞	4	n/a
10	16	∞	4	n/a

Tentative distance: ∞
Up-down path root: n/a

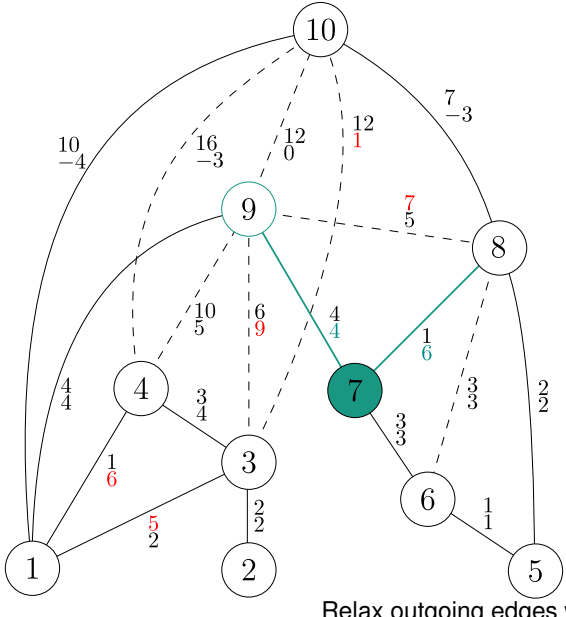
Update target to its parent



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	∞	4	n/a
10	16	∞	4	n/a

Tentative distance: ∞
Up-down path root: n/a

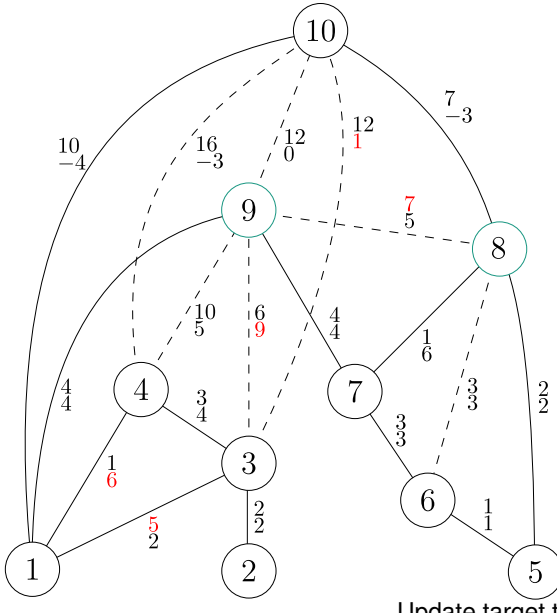
Target < source, so relax backward from target



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	7	4	7
10	16	∞	4	n/a

Tentative distance: 17
Up-down path root: 9

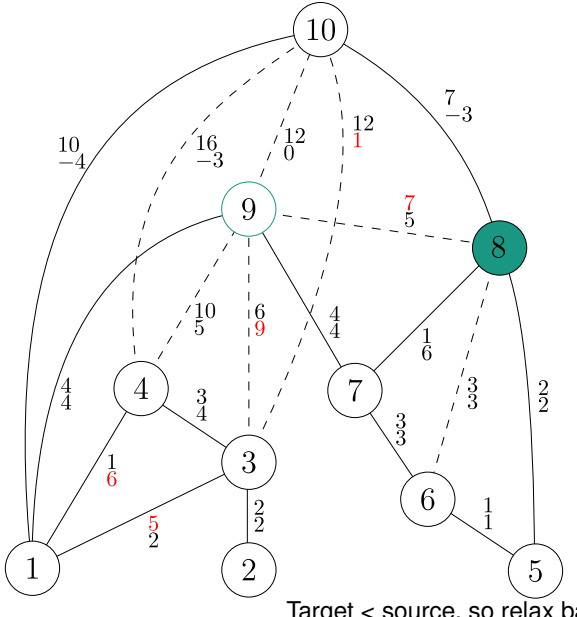
Relax outgoing edges with downweights



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	7	4	7
10	16	∞	4	n/a

Tentative distance: 17
Up-down path root: 9

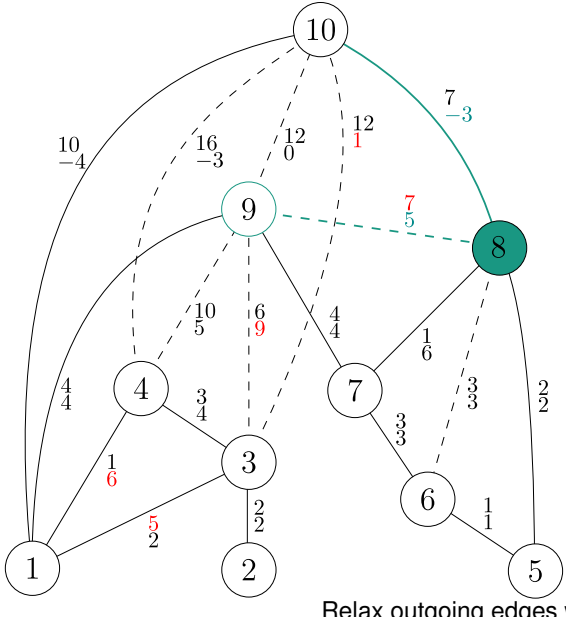
Update target to its parent



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	7	4	7
10	16	∞	4	n/a

Tentative distance: 17
Up-down path root: 9

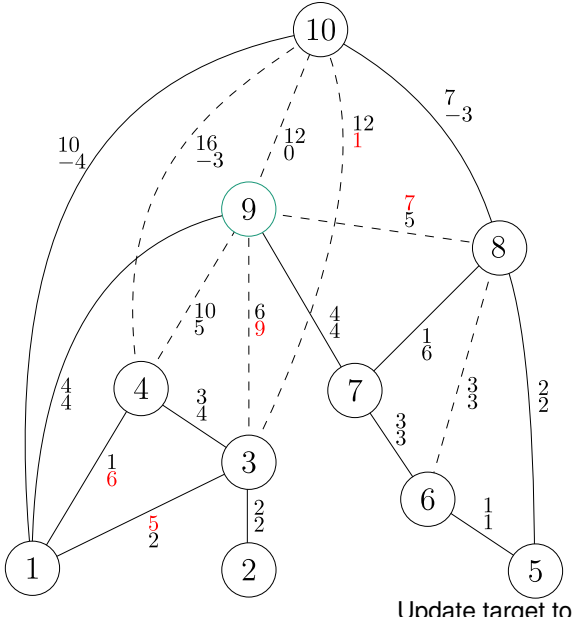
Target < source, so relax backward from target



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	7	4	7
10	16	0	4	8

Tentative distance: 16
Up-down path root: 10

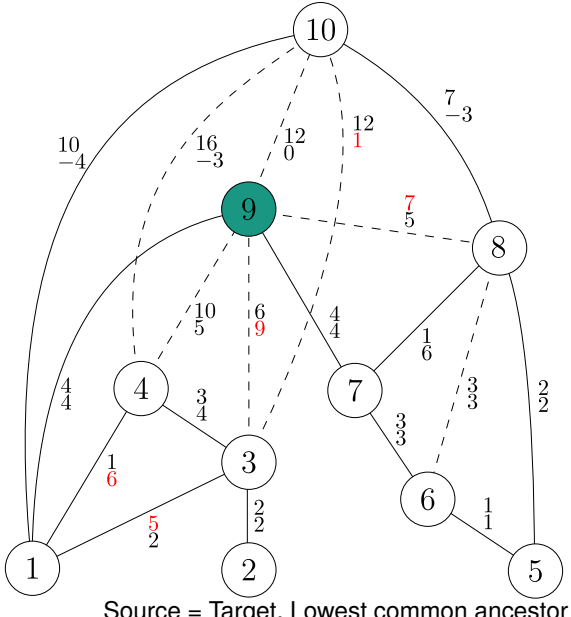
Relax outgoing edges with downweights



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	7	4	7
10	16	0	4	8

Tentative distance: 16
Up-down path root: 10

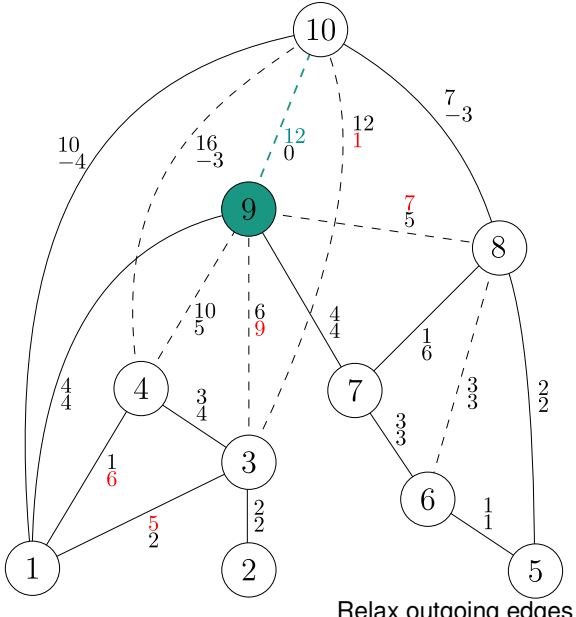
Update target to its parent



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	7	4	7
10	16	0	4	8

Tentative distance: 16
Up-down path root: 10

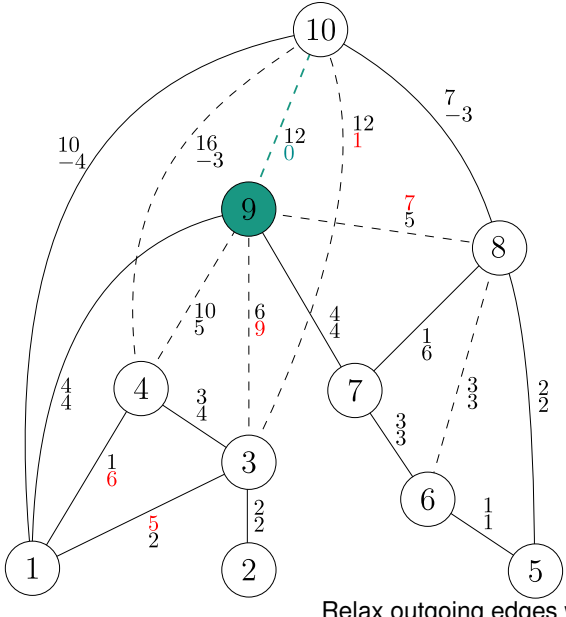
Source = Target, Lowest common ancestor found so relax forward and backward



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	7	4	7
10	16	0	4	8

Tentative distance: 16
Up-down path root: 10

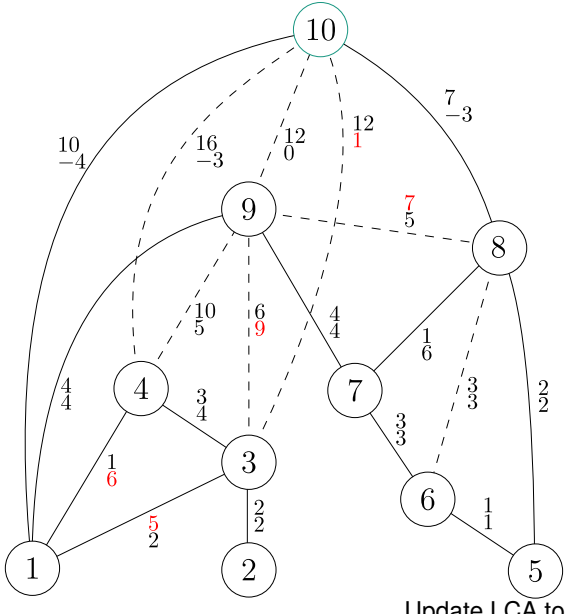
Relax outgoing edges with upweights



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	7	4	7
10	16	0	4	8

Tentative distance: 16
Up-down path root: 10

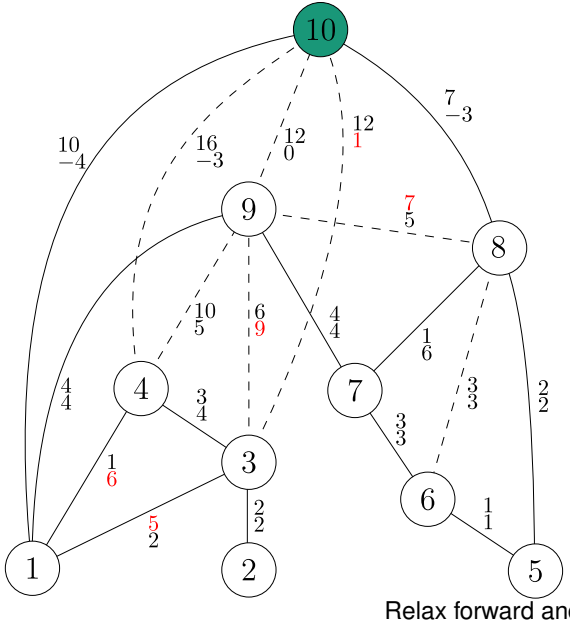
Relax outgoing edges with downweights



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	7	4	7
10	16	0	4	8

Tentative distance: 16
Up-down path root: 10

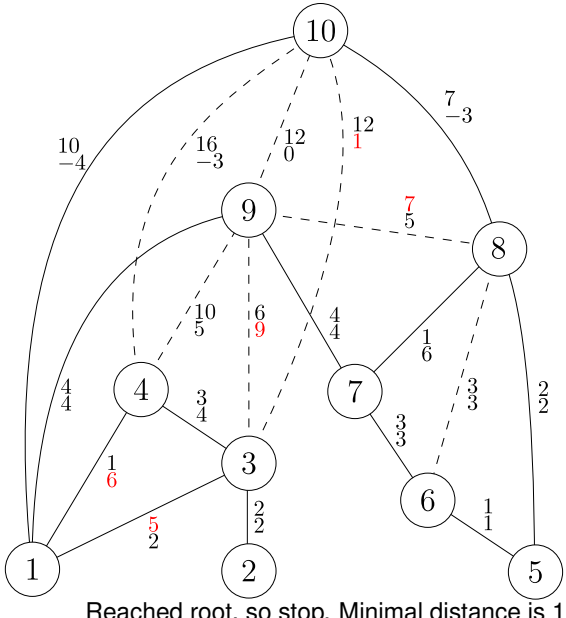
Update LCA to its parent



Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	7	4	7
10	16	0	4	8

Tentative distance: 16
Up-down path root: 10

Relax forward and backward

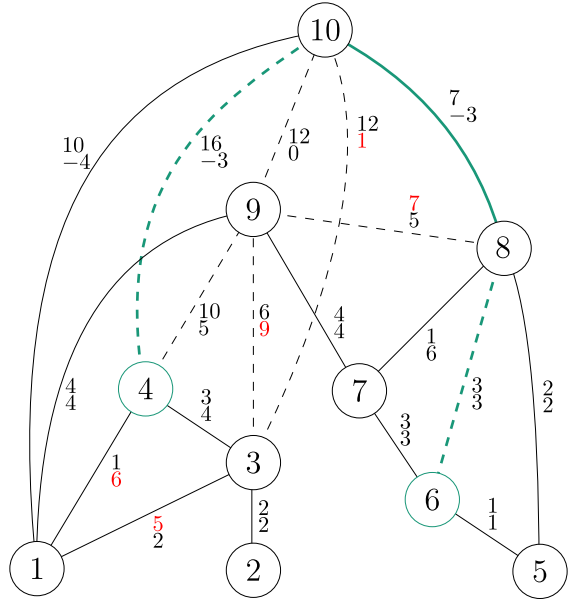


Vertex	d_f	d_b	p_f	p_b
1	∞	∞	n/a	n/a
2	∞	∞	n/a	n/a
3	∞	∞	n/a	n/a
4	0	∞	4	n/a
5	∞	∞	n/a	n/a
6	∞	0	n/a	6
7	∞	3	n/a	6
8	∞	3	n/a	6
9	10	7	4	7
10	16	0	4	8

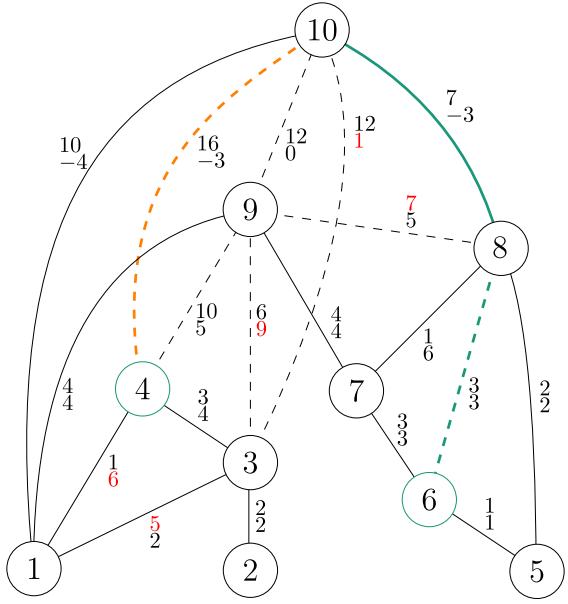
Tentative distance: 16
Up-down path root: 10

Reached root, so stop. Minimal distance is 16 with up-down path: $4 \to 10 \to 8 \to 6$

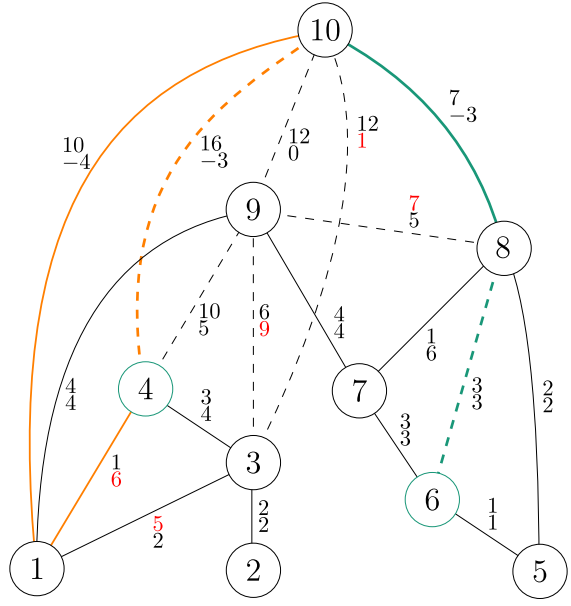
Path Retrieval



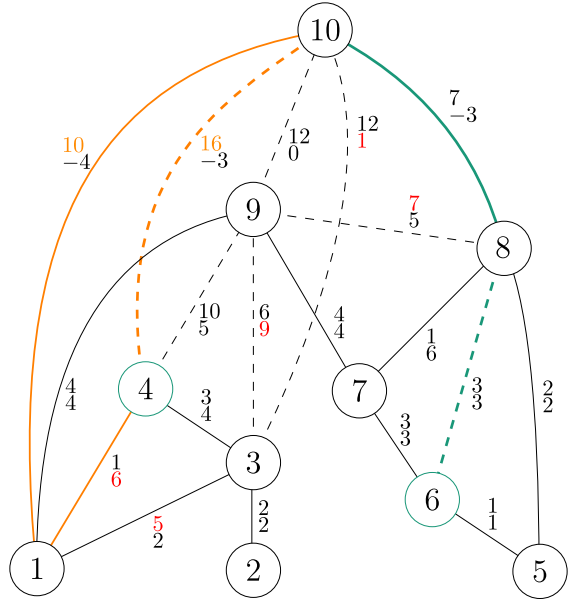
Recursively unpack up-down path consisting of shortcuts $(4, 10)_u, (8, 10)_d, (6, 8)_d$



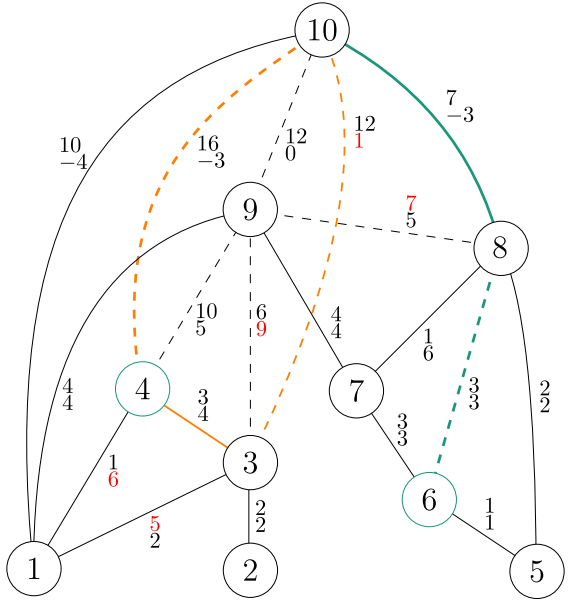
Unpack upward shortcut $(4, 10)_u$



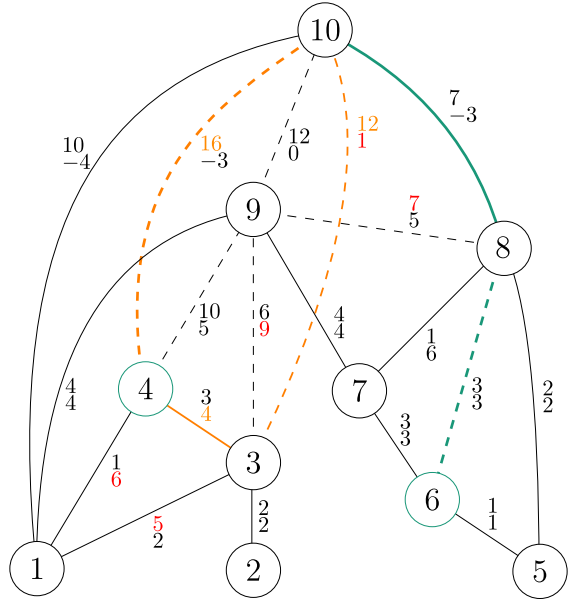
$\{1, 4, 10\}$ is a lower triangle of $(4, 10)_u$



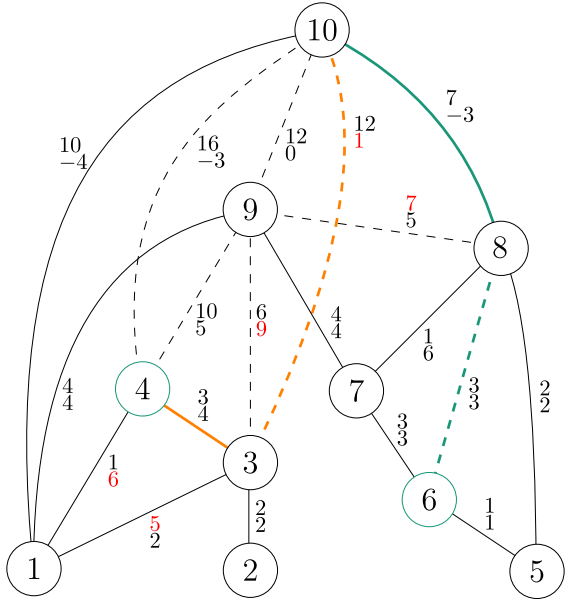
Check for triangle equality: false, because $(1,4)_d$ was deleted



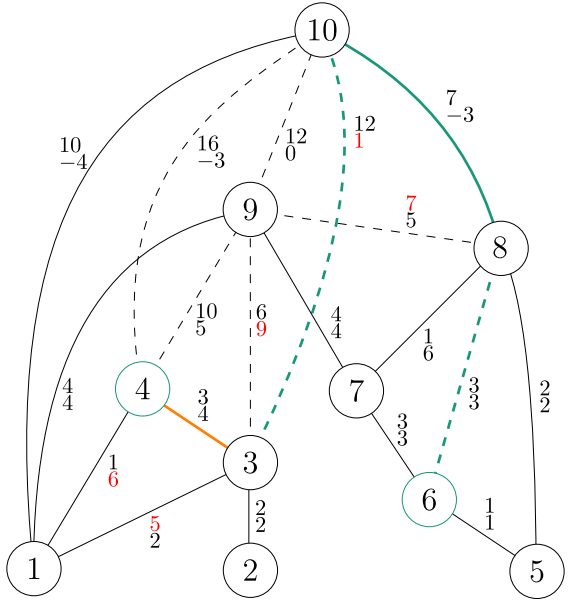
$\{3, 4, 10\}$ is a lower triangle of $(4, 10)_u$



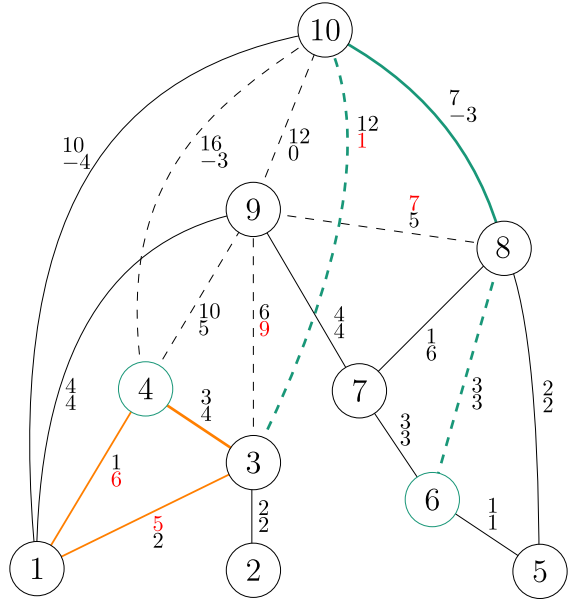
Check for triangle equality: true



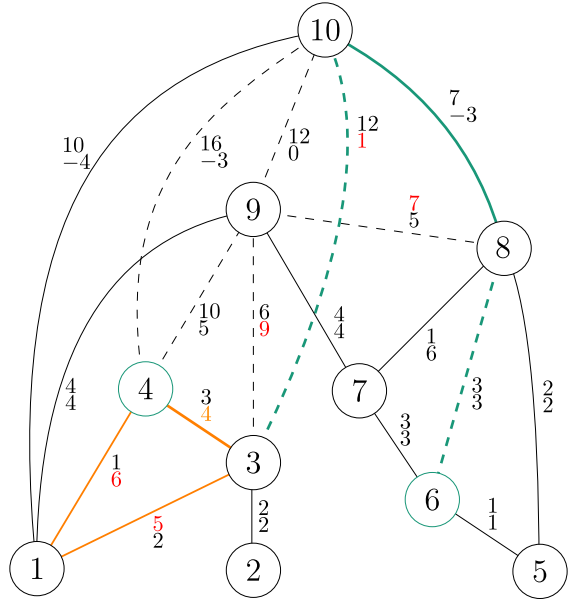
Substitute $(4, 10)_u$ by $(3, 4)_d$ and $(3, 10)_u$ and unpack recursively



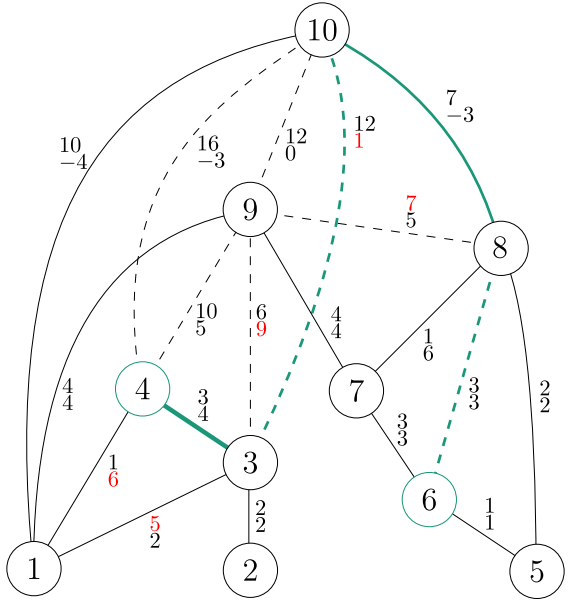
Unpack downward shortcut $(3, 4)_d$



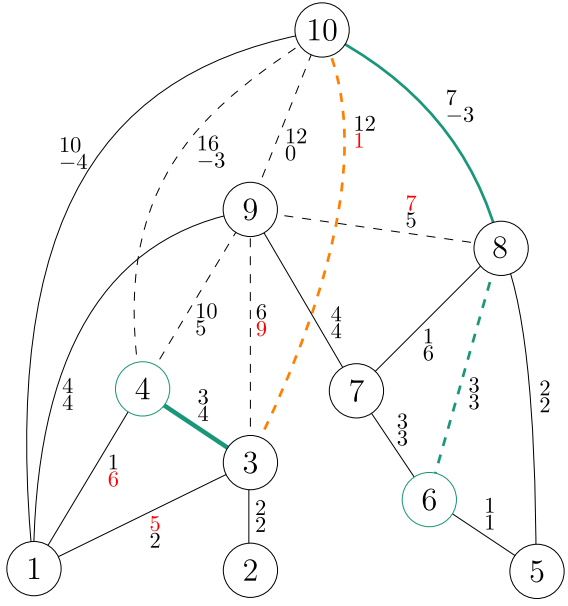
$\{1, 3, 4\}$ is a lower triangle of $(3, 4)_d$



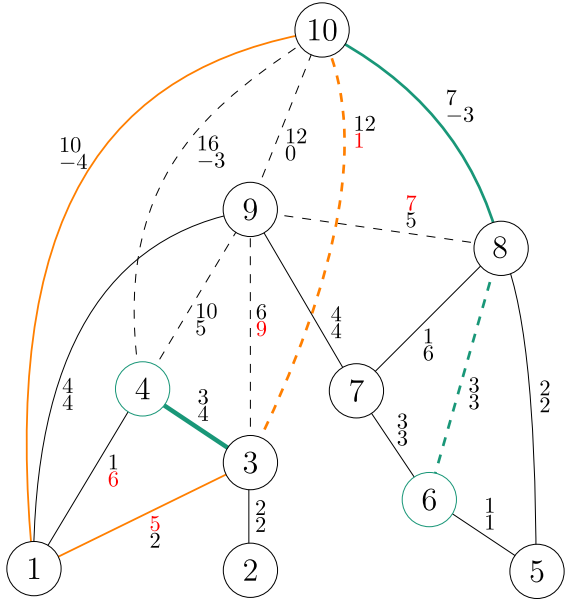
Check for triangle equality: false, because $(1,4)_d$ and $(1,3)_u$ were deleted



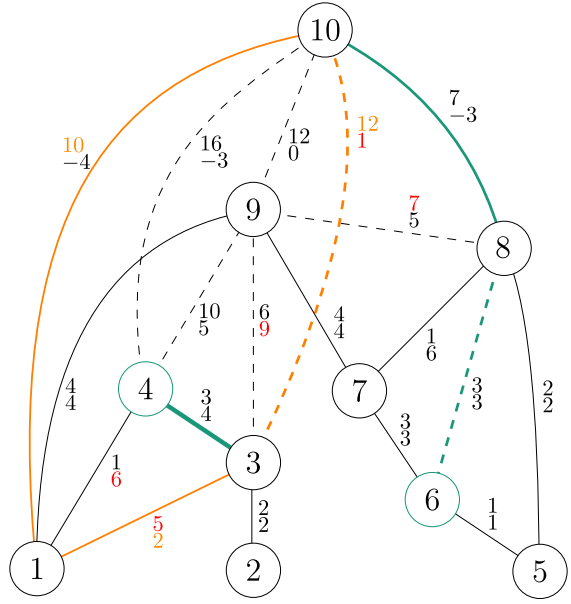
No more lower triangles $\rightarrow (3,4)_d$ is an original edge



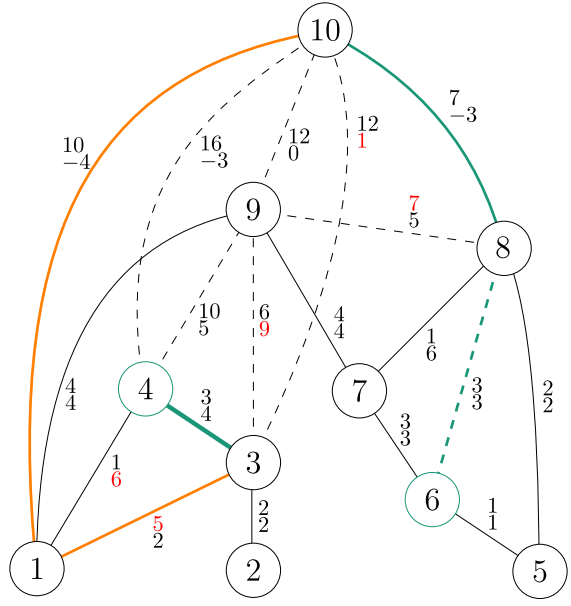
Unpack upward shortcut $(3, 10)_u$



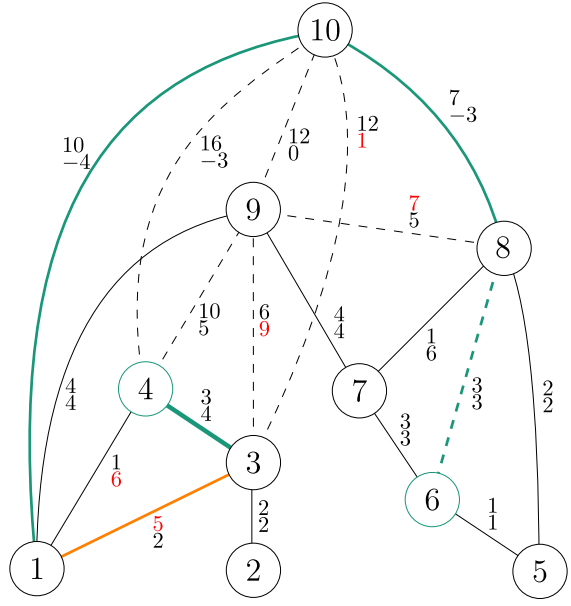
$\{1, 3, 10\}$ is a lower triangle of $(3, 10)_u$



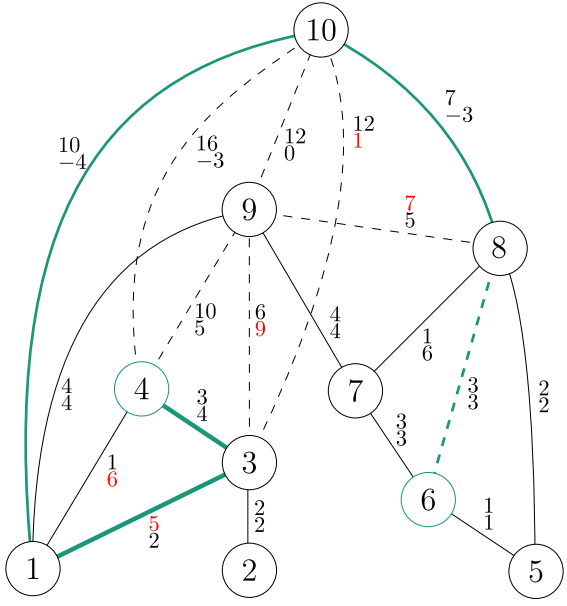
Check for triangle equality: true



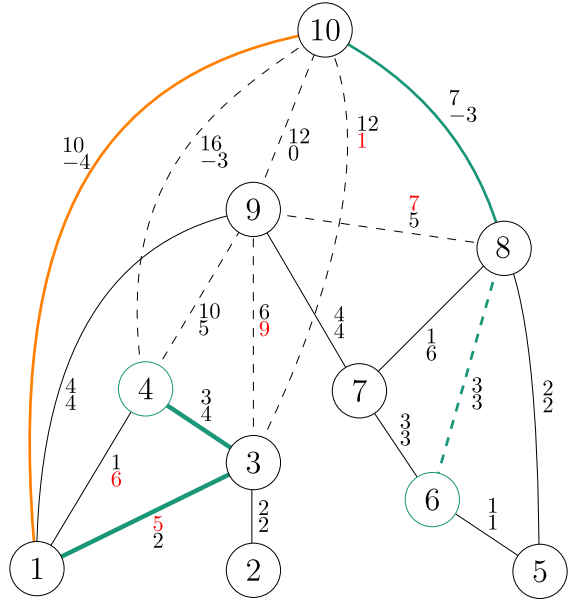
Substitute $(3, 10)_u$ by $(1, 3)_d$ and $(1, 10)_u$ and unpack recursively



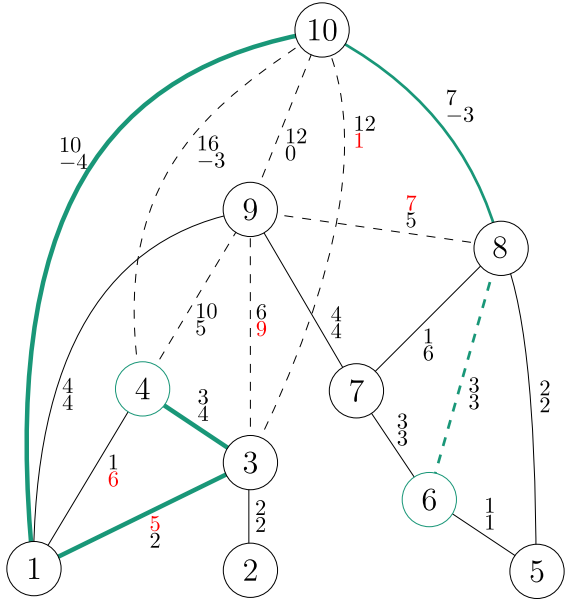
Unpack downward shortcut $(1, 3)_d$



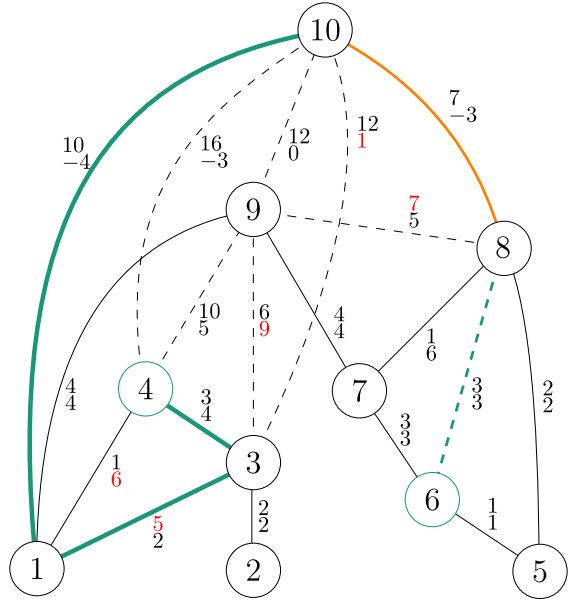
No more lower triangles $\rightarrow (1,3)_d$ is an original edge



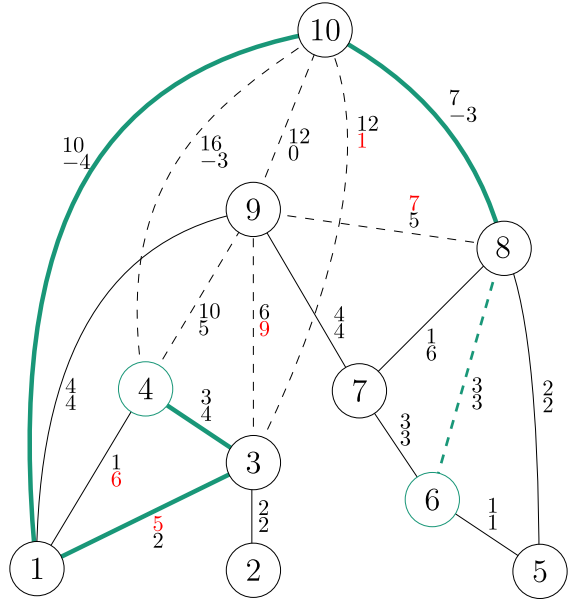
Unpack upward shortcut $(1, 10)_u$



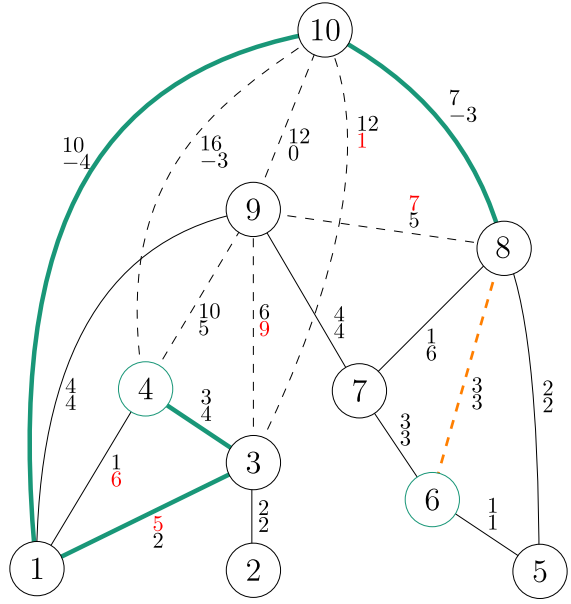
No more lower triangles $\rightarrow (1, 10)_u$ is an original edge



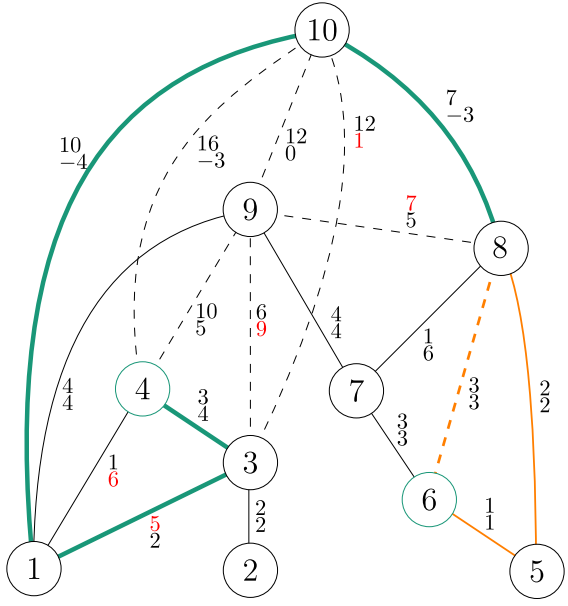
Unpack downward shortcut $(8, 10)_d$



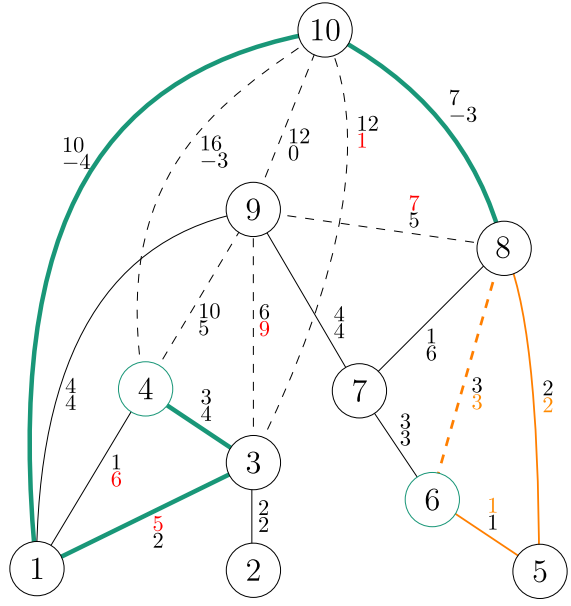
No more lower triangles $\rightarrow (8, 10)_d$ is an original edge



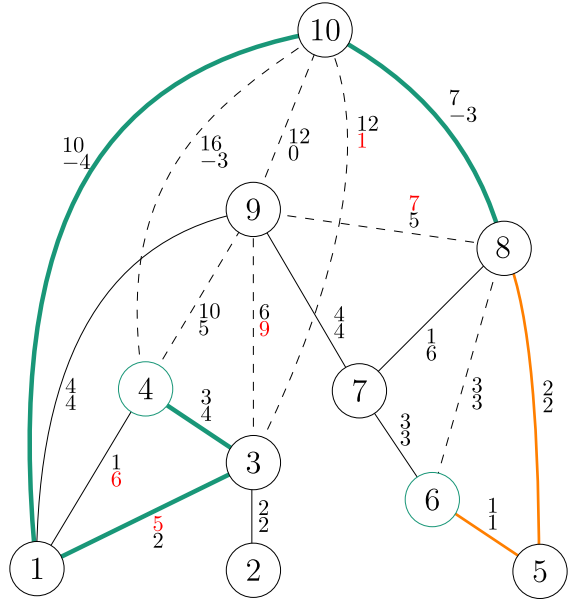
Unpack downward shortcut $(6, 8)_d$



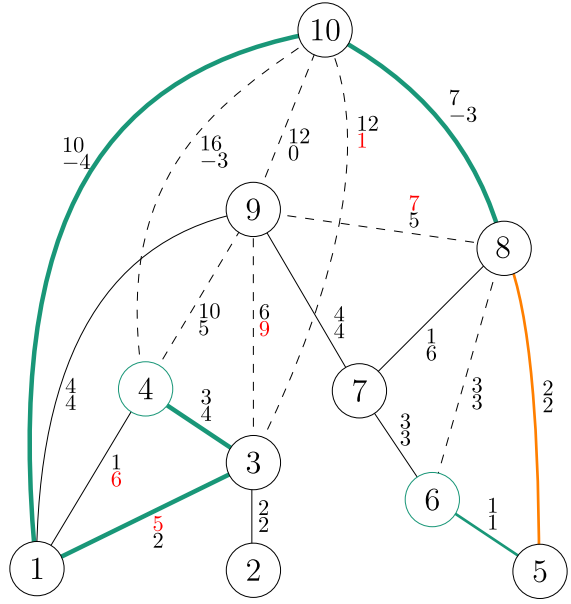
$\{5, 6, 8\}$ is a lower triangle of $(6, 8)_d$



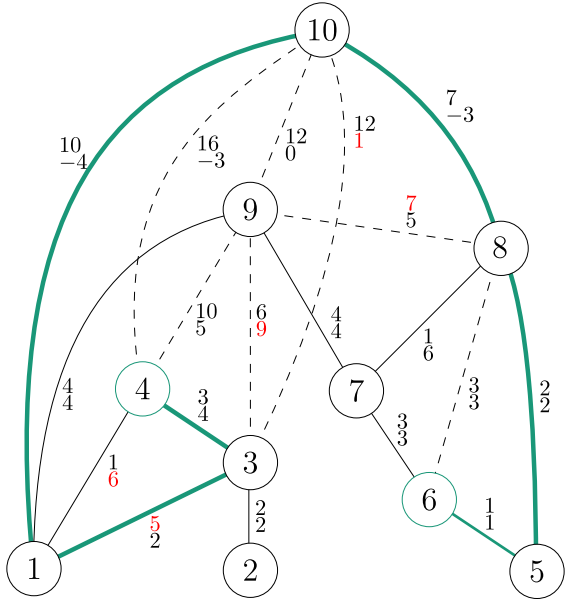
Check for triangle equality: true



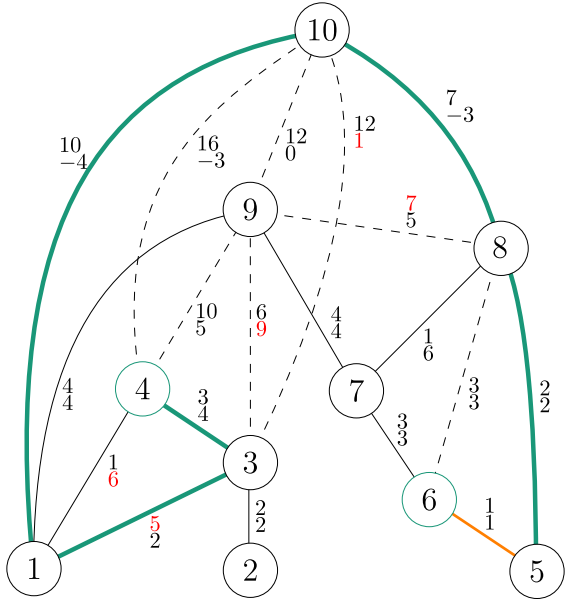
Substitute $(6, 8)_d$ by $(5, 8)_d$ and $(5, 6)_u$ and unpack recursively



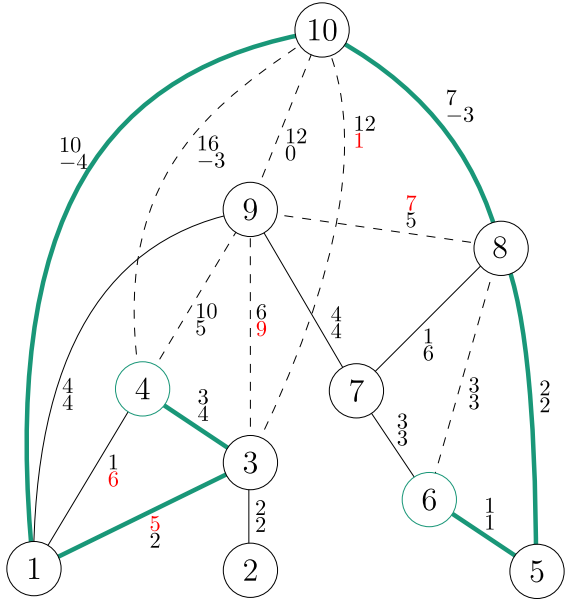
Unpack downward shortcut $(5, 8)_d$



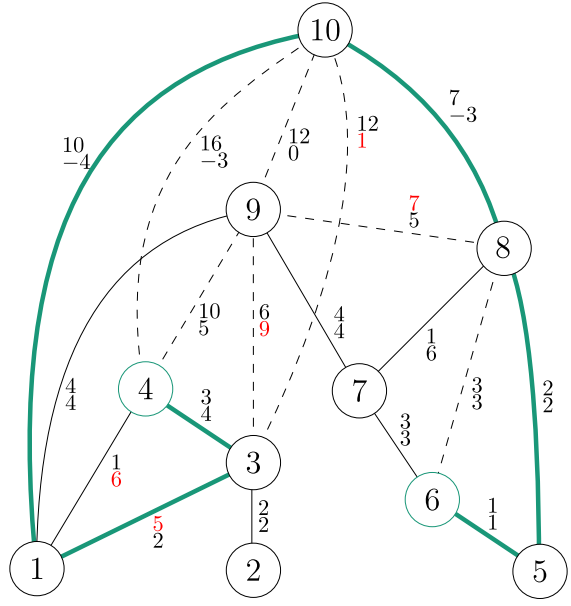
No more lower triangles $\rightarrow (5,8)_d$ is an original edge



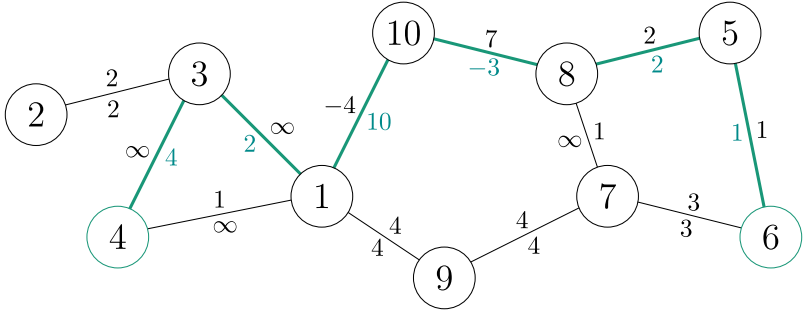
Unpack upward shortcut $(5, 6)_u$



No more lower triangles $\rightarrow (5,6)_u$ is an original edge



Final path is 4 → 3 → 1 → 10 → 8 → 5 → 6



Final path on the original graph