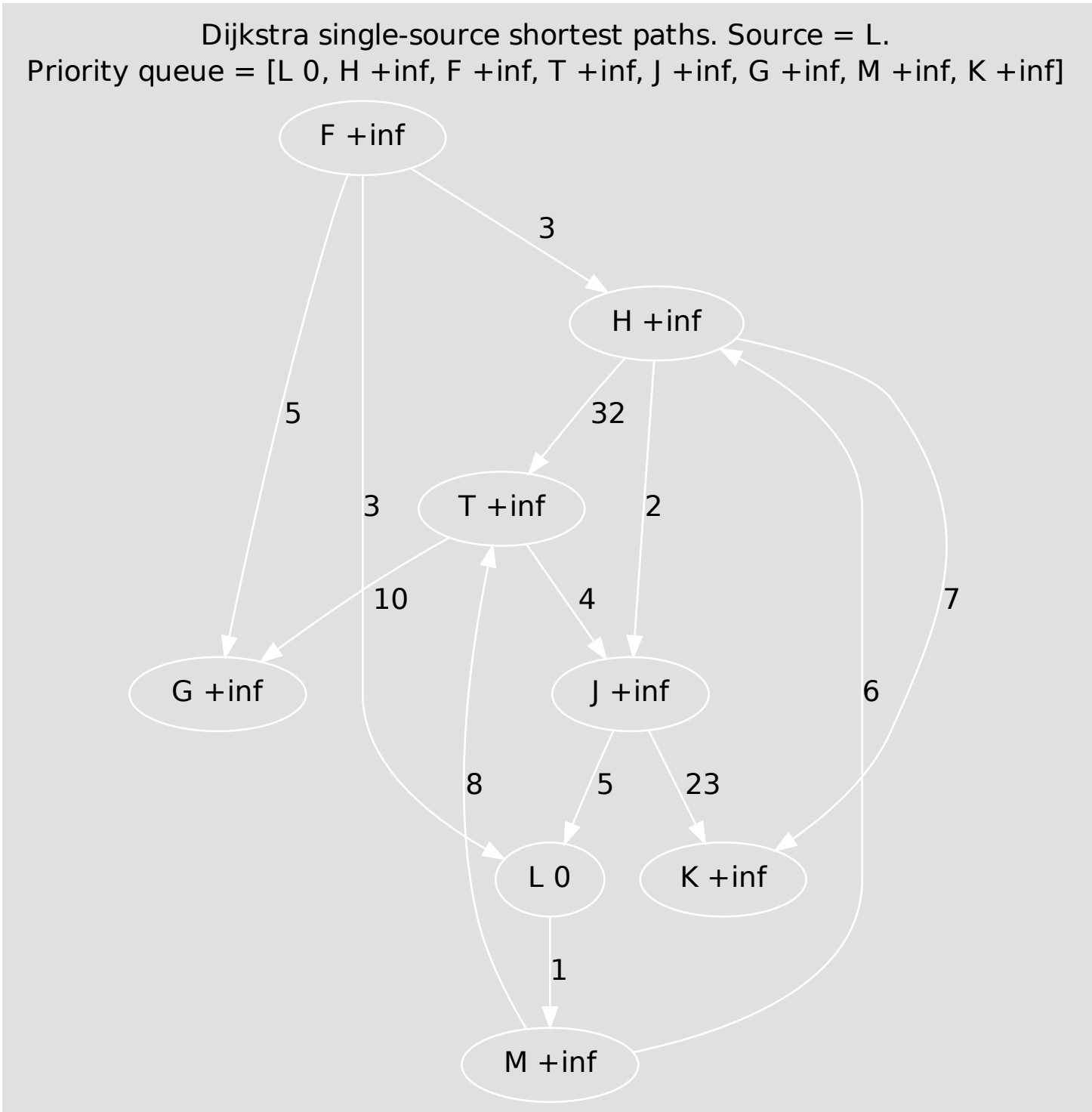
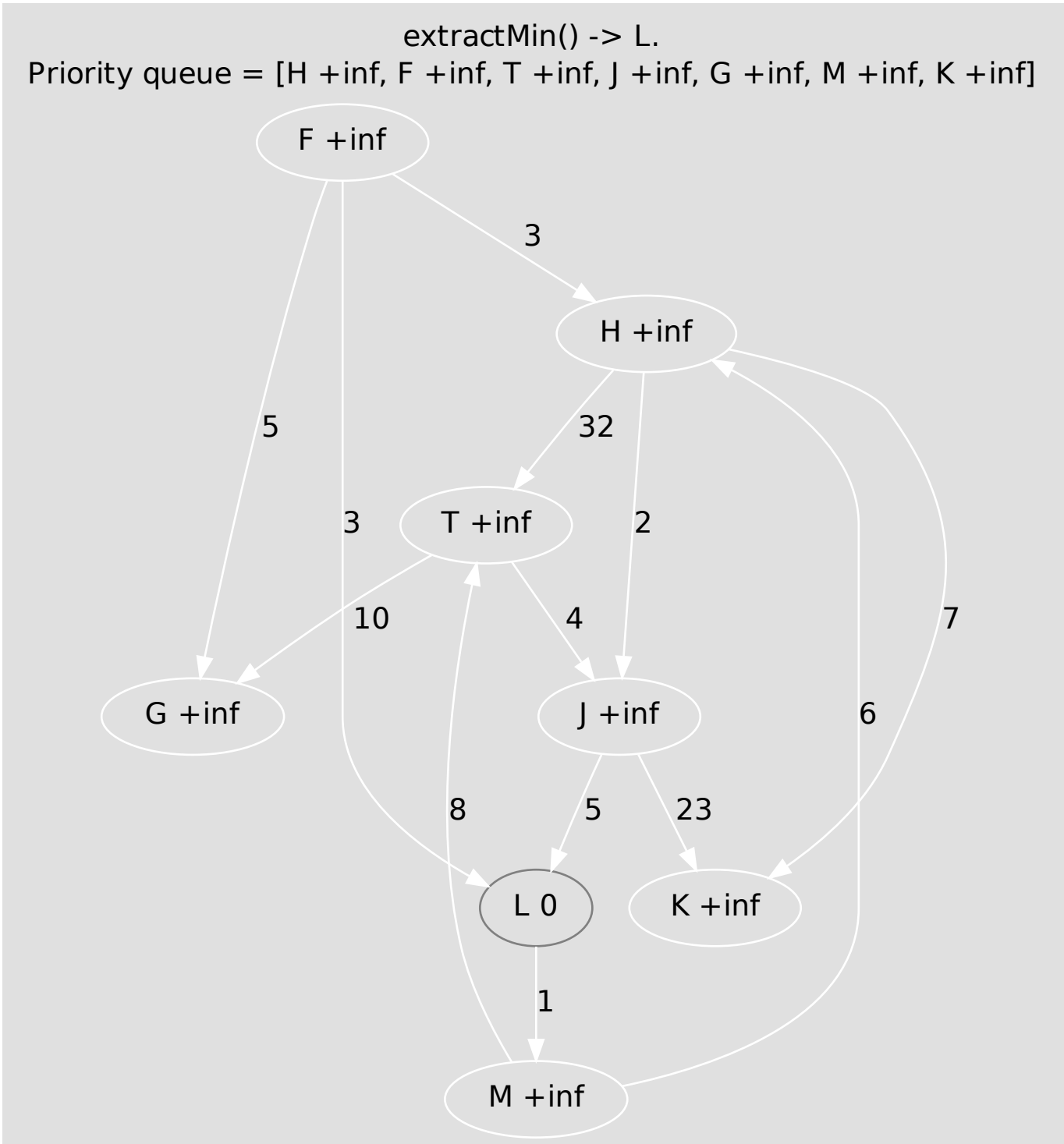
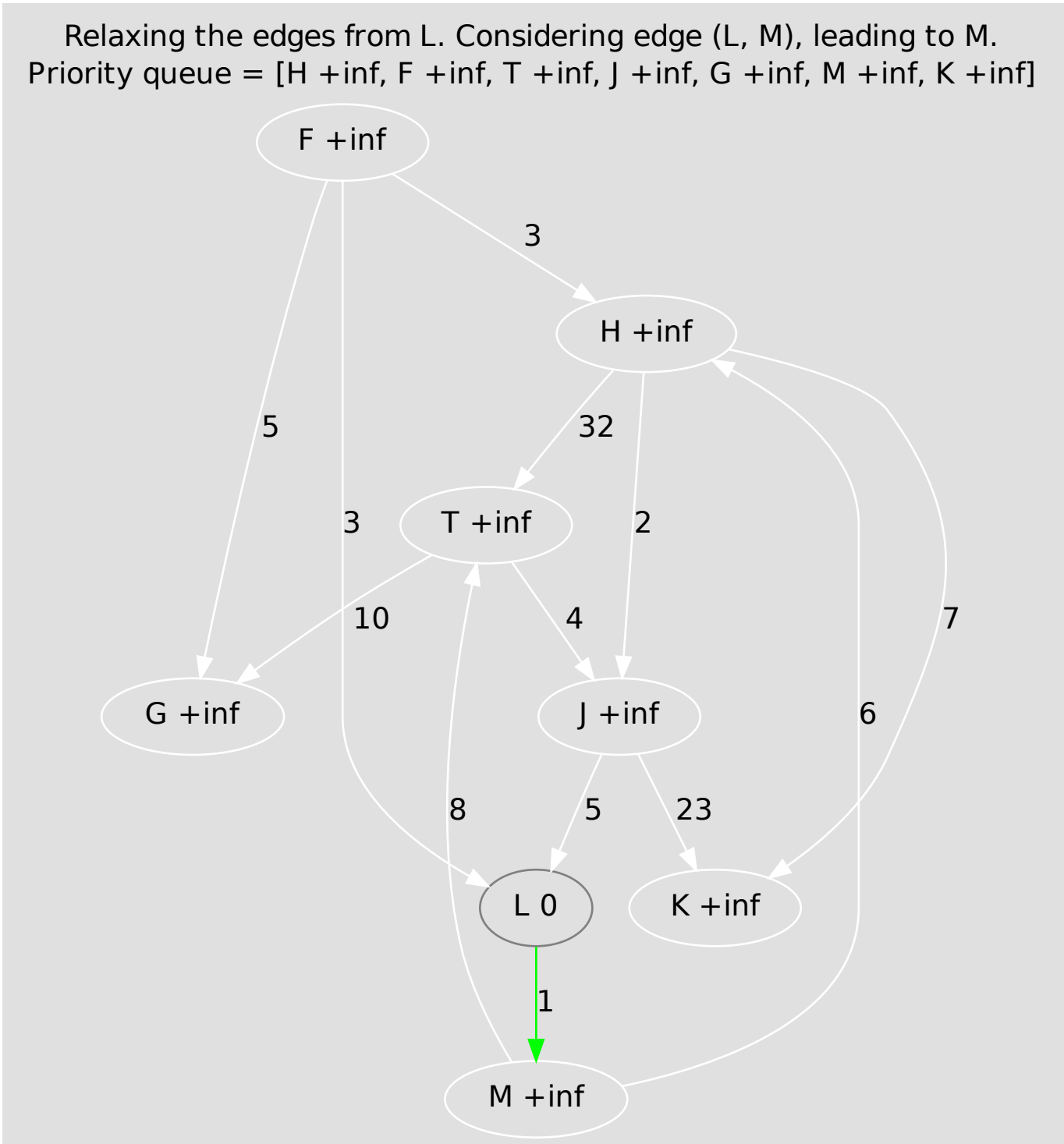
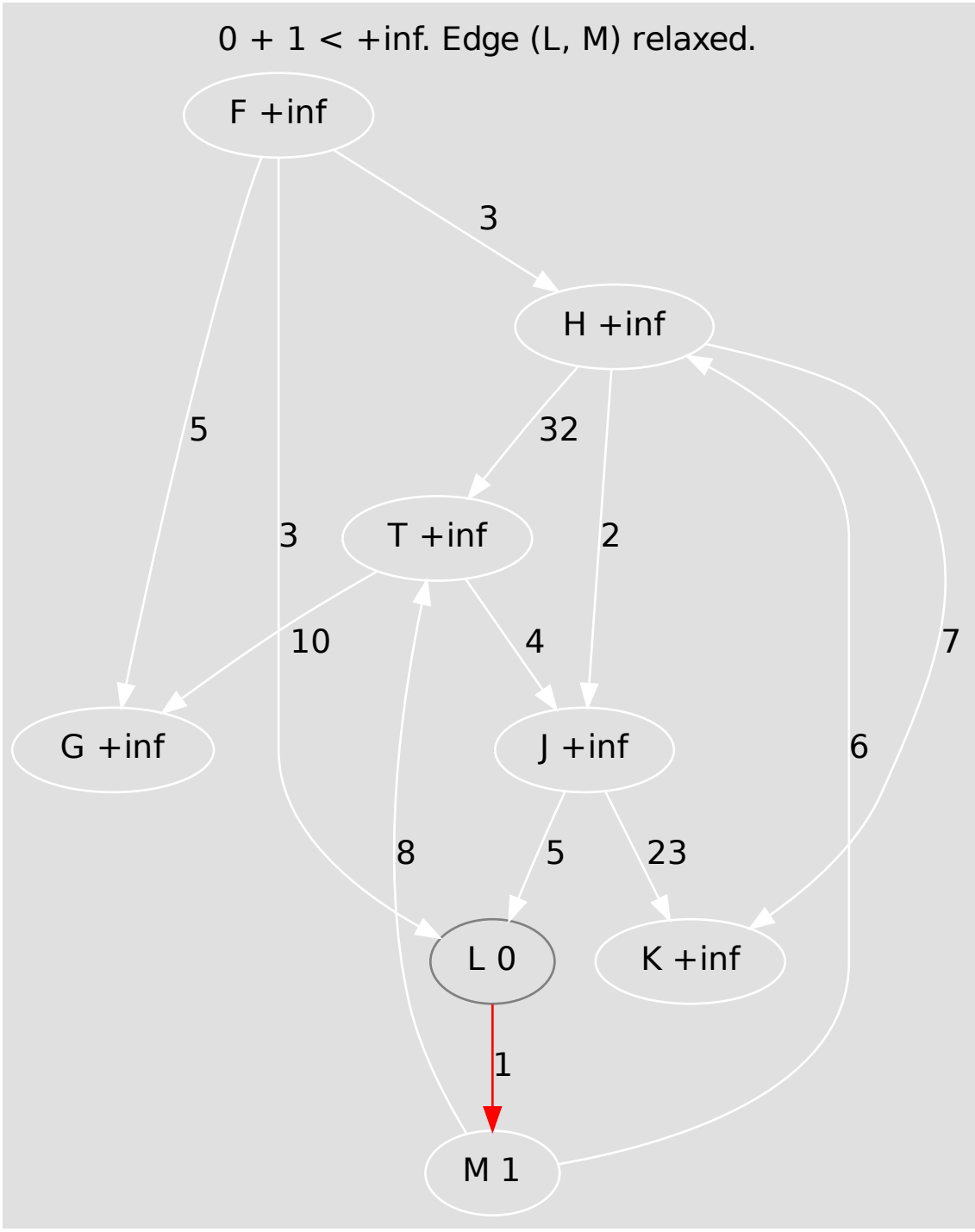
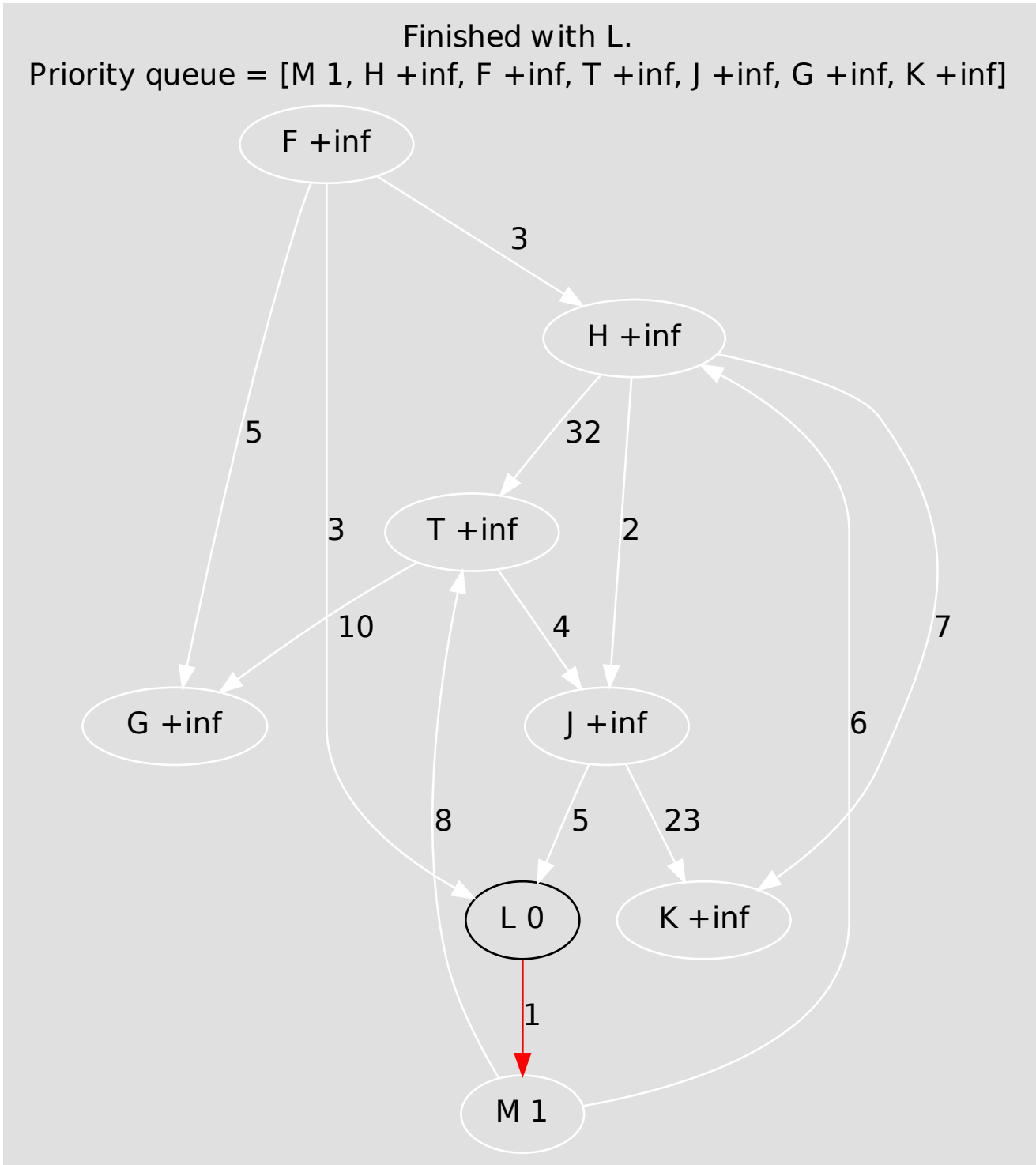
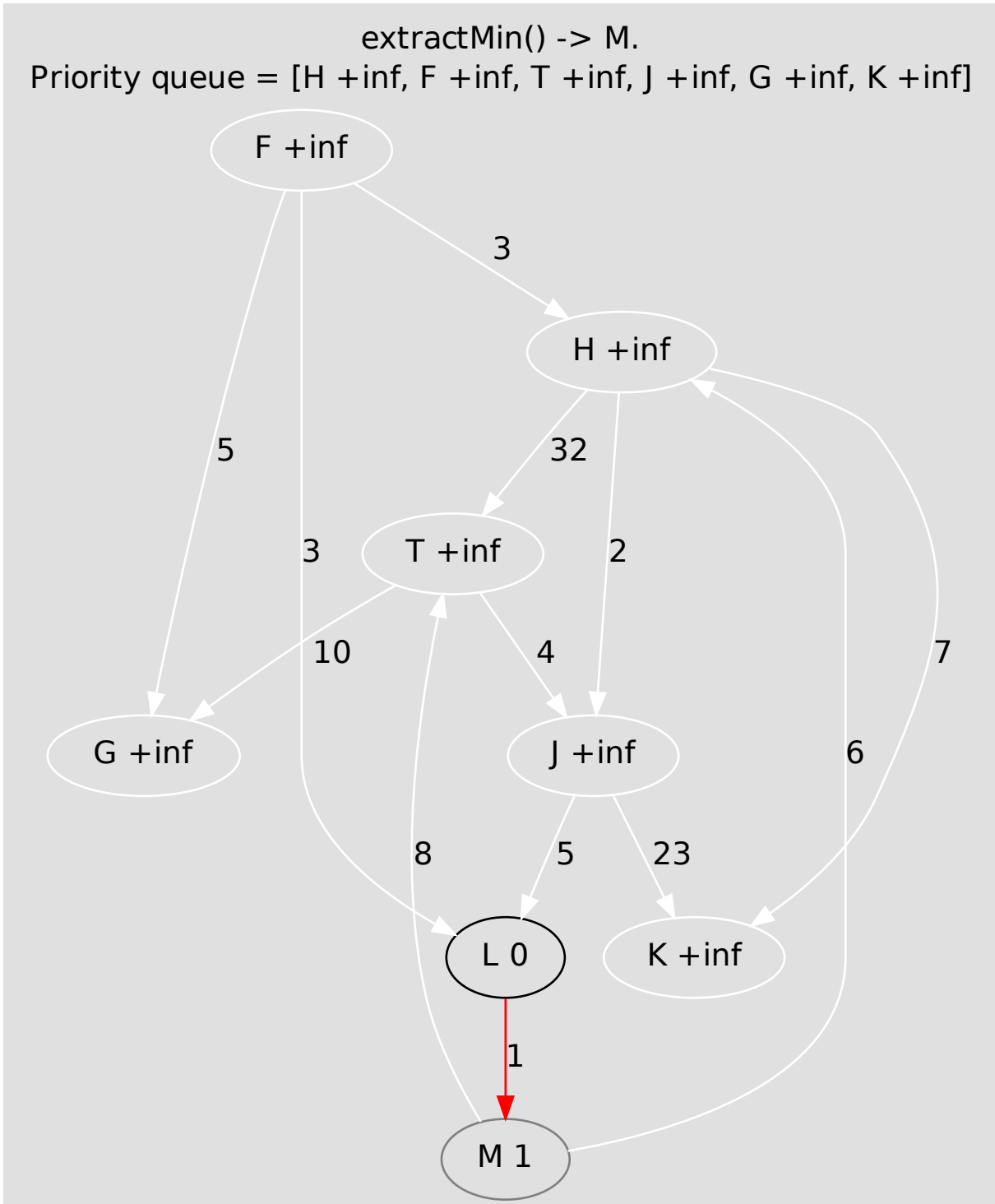

Dijkstra single-source shortest paths. Source = L.
Priority queue = [L 0, H +inf, F +inf, T +inf, J +inf, G +inf, M +inf, K +inf]
F +inf
3
H +inf
5
32
3
T +inf
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M +inf

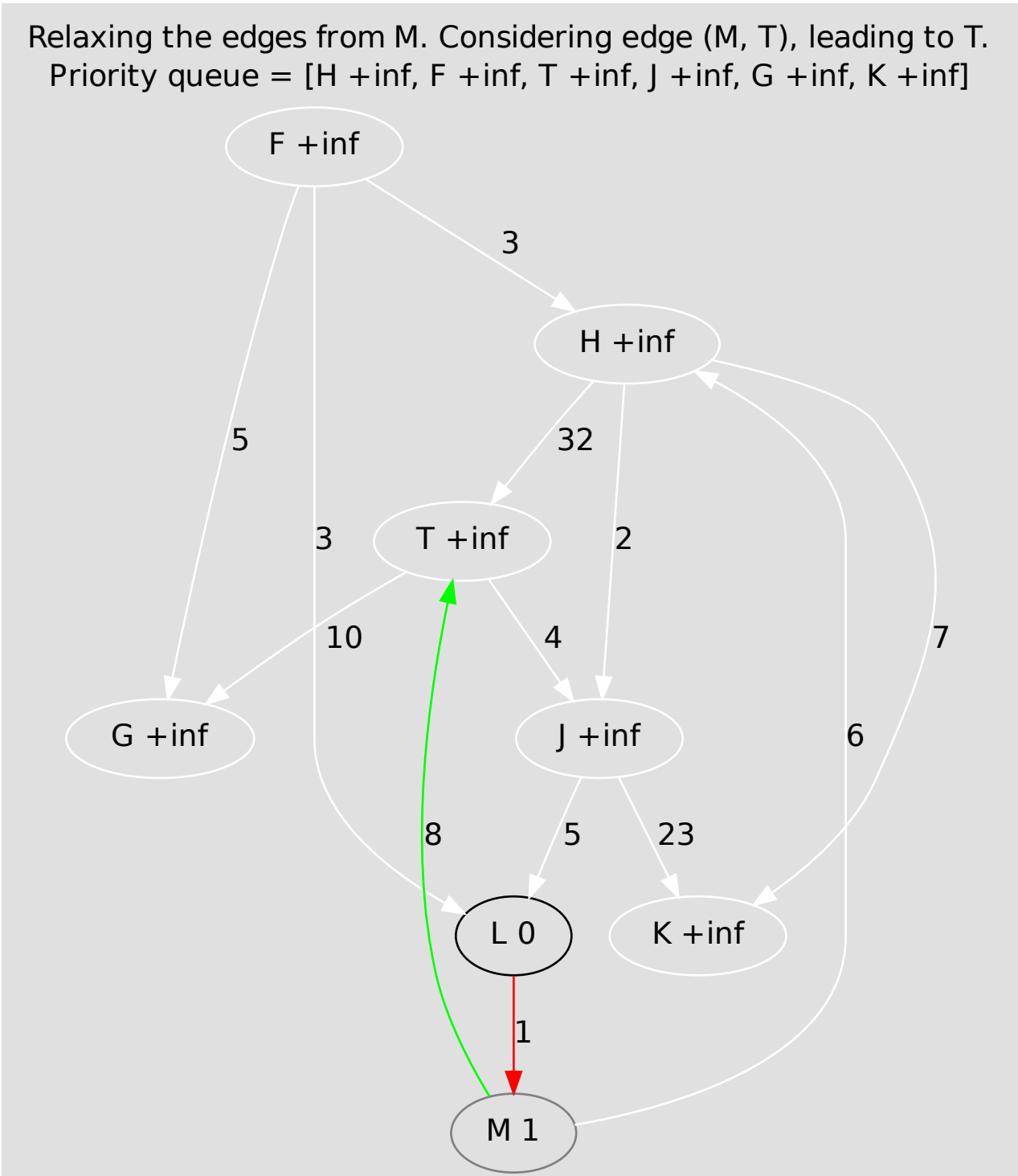

extractMin() -> L.
Priority queue = [H +inf, F +inf, T +inf, J +inf, G +inf, M +inf, K +inf]
F +inf
3
H +inf
5
32
3
T +inf
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M +inf

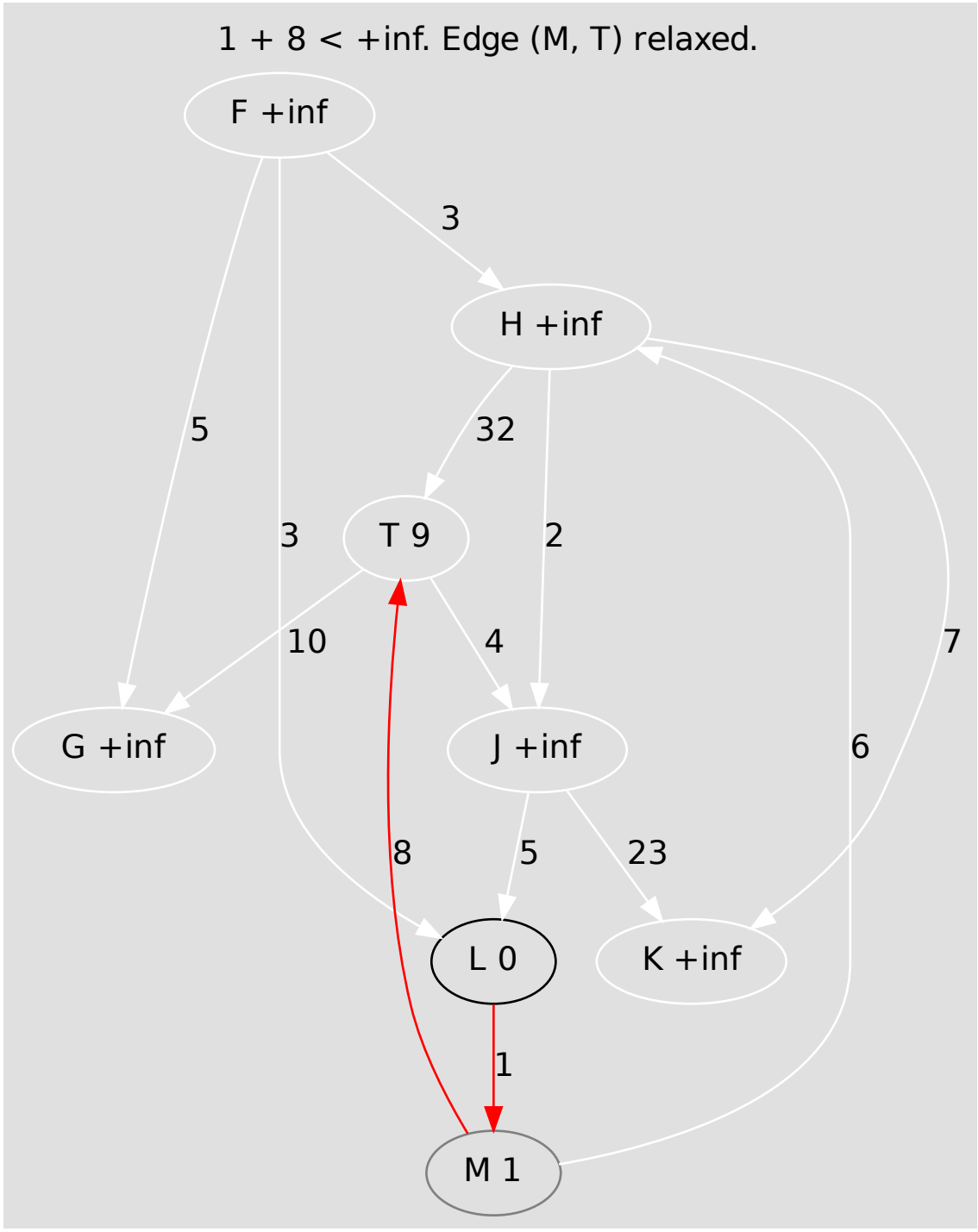

Relaxing the edges from L. Considering edge (L, M), leading to M.
Priority queue = [H +inf, F +inf, T +inf, J +inf, G +inf, M +inf, K +inf]
F +inf
3
H +inf
5
32
3
T +inf
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M +inf

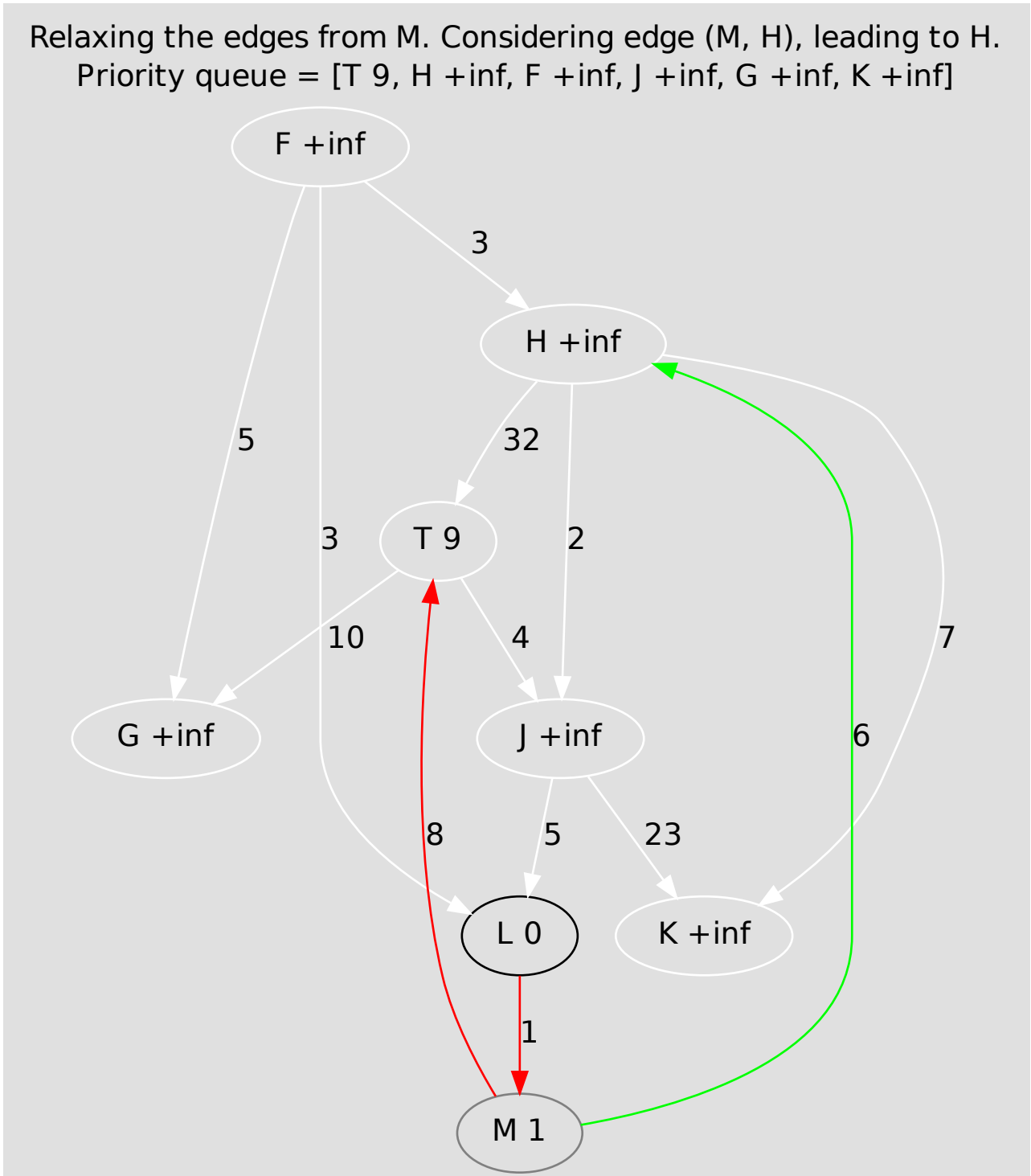

0 + 1 < +inf. Edge (L, M) relaxed.
F +inf
3
H +inf
5
32
3
T +inf
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M 1

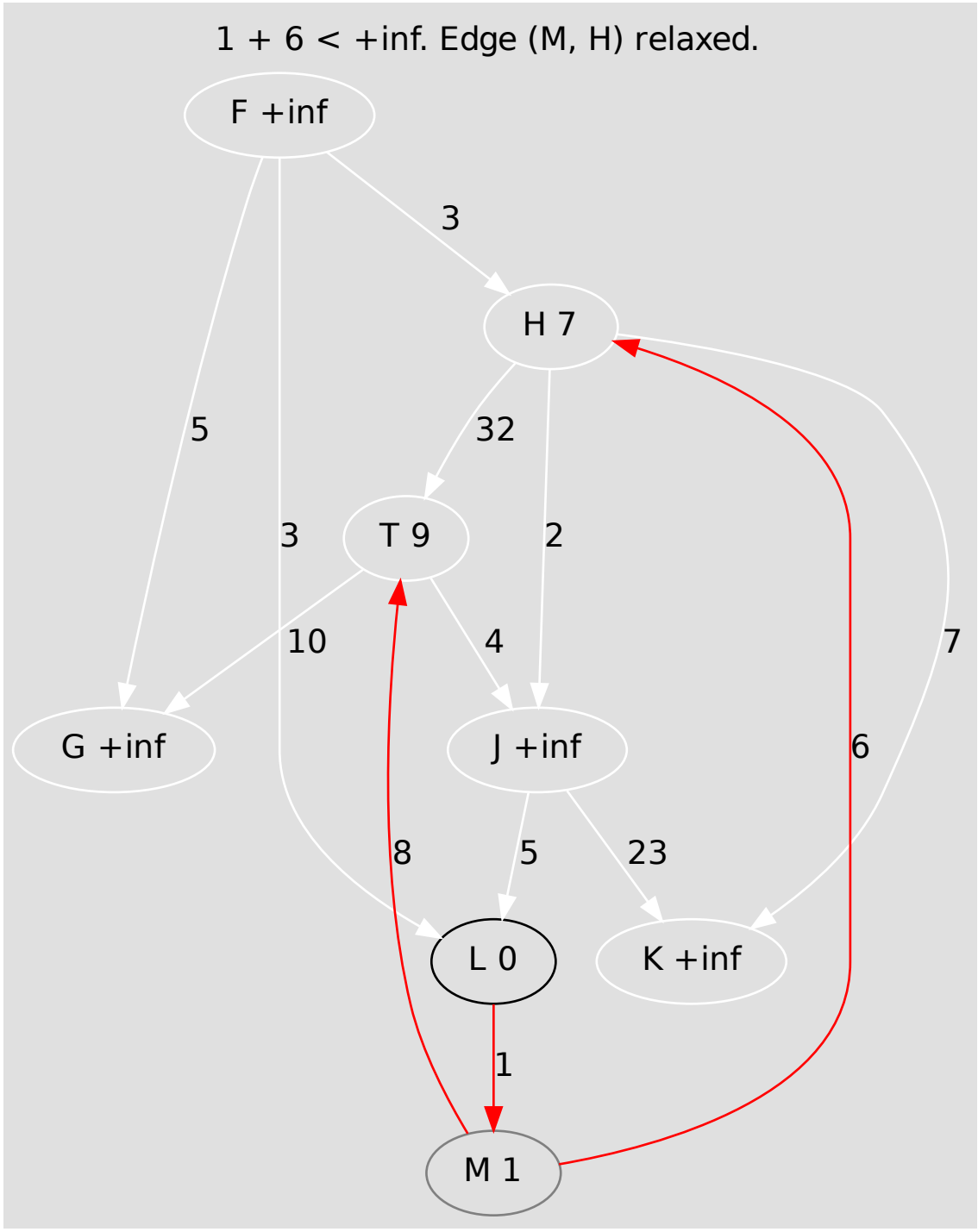

Finished with L.
Priority queue = [M 1, H +inf, F +inf, T +inf, J +inf, G +inf, K +inf]
F +inf
3
H +inf
5
32
3
T +inf
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M 1

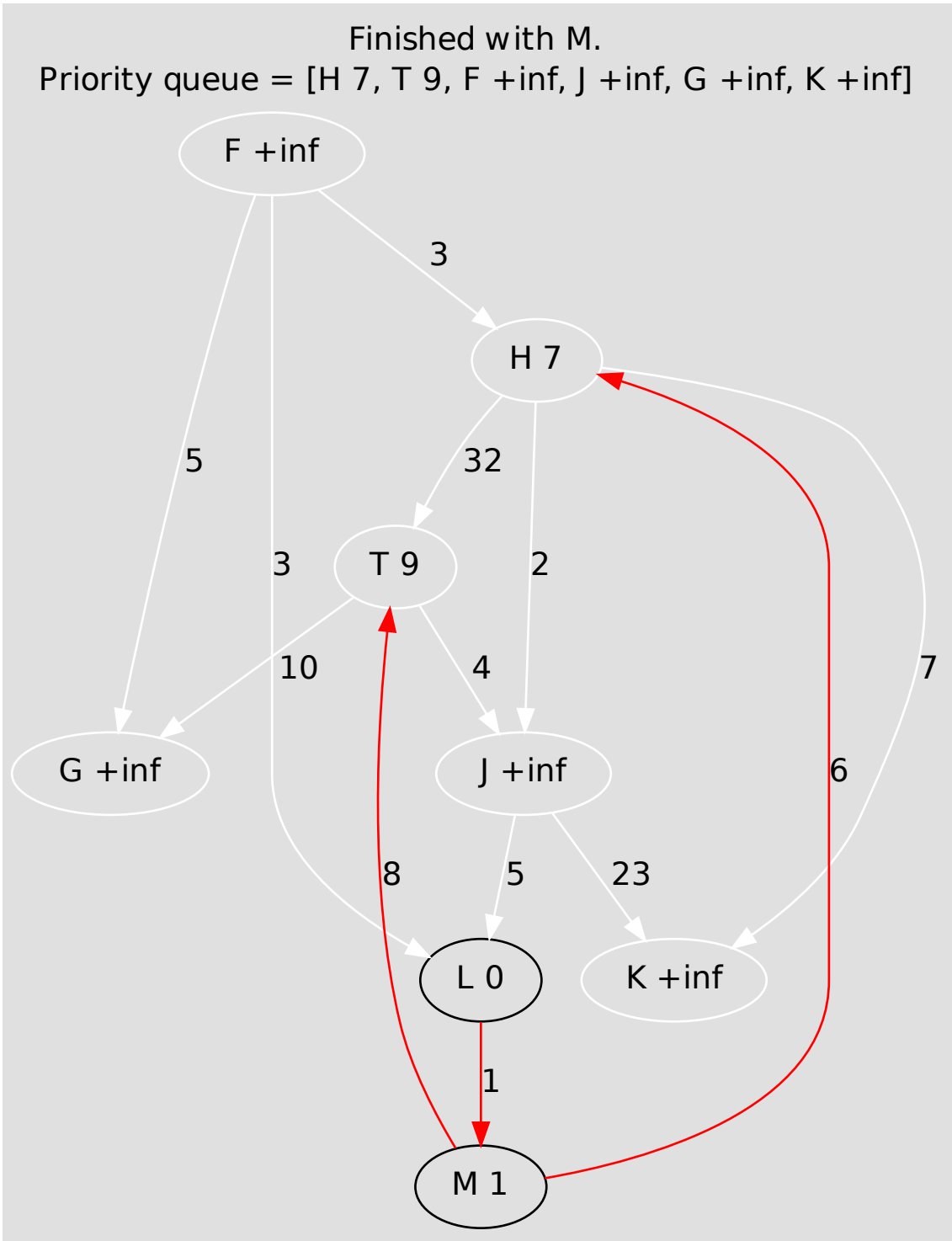

extractMin() -> M.
Priority queue = [H +inf, F +inf, T +inf, J +inf, G +inf, K +inf]
F +inf
3
H +inf
5
32
3
T +inf
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M 1

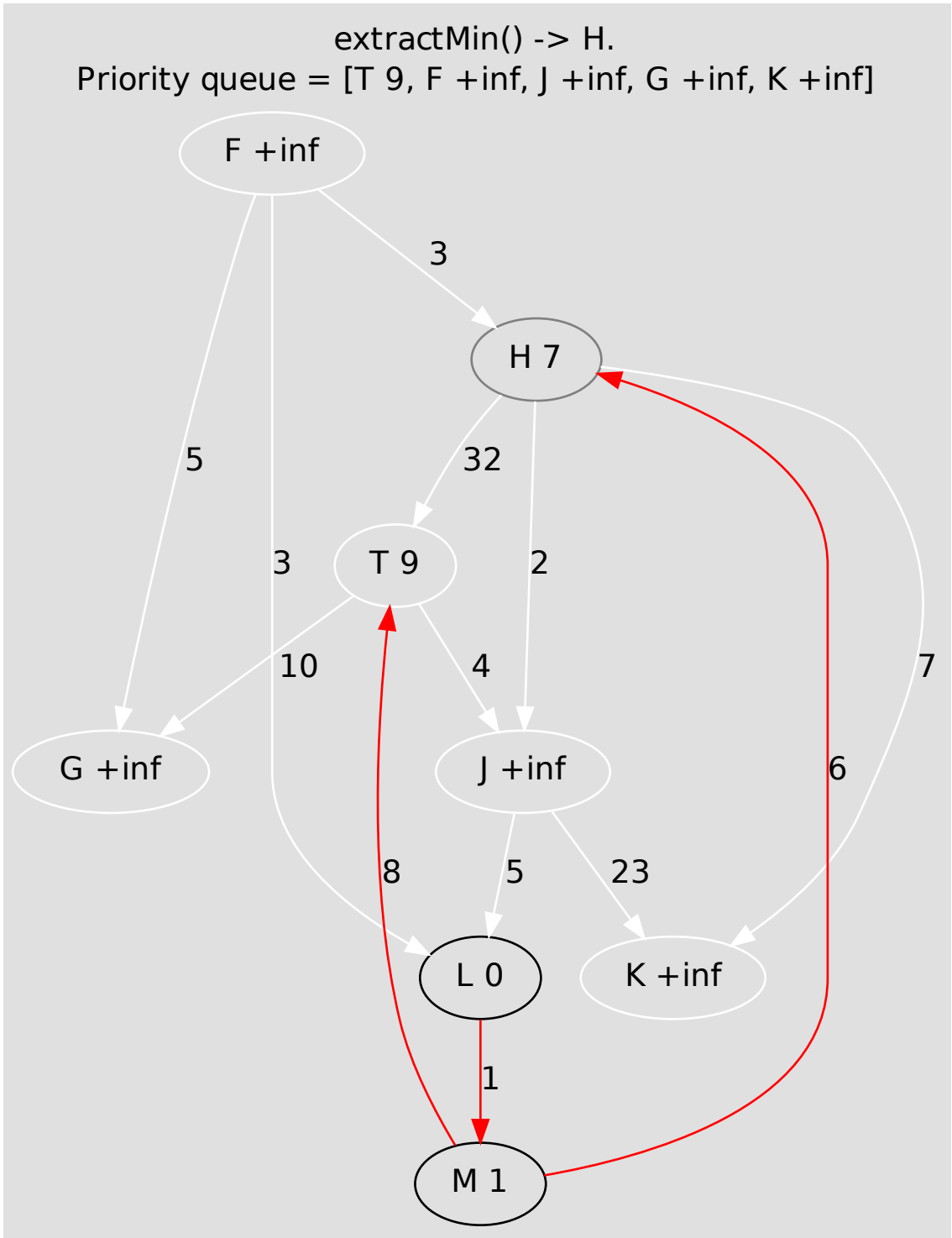

Relaxing the edges from M. Considering edge (M, T), leading to T.
Priority queue = [H +inf, F +inf, T +inf, J +inf, G +inf, K +inf]
F +inf
3
H +inf
5
32
3
T +inf
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M 1

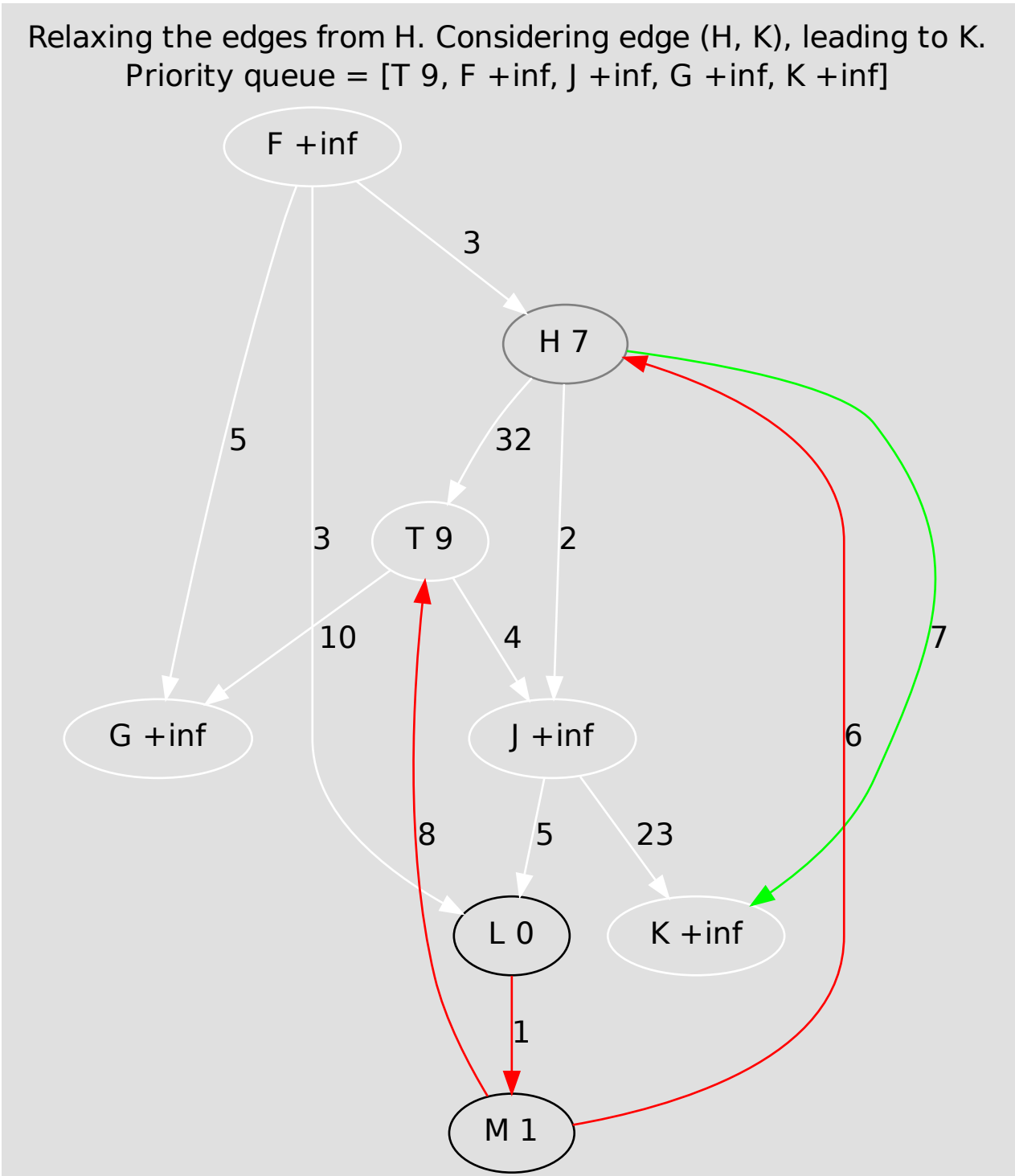

1 + 8 < +inf. Edge (M, T) relaxed.
F +inf
3
H +inf
5
32
3
T 9
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M 1

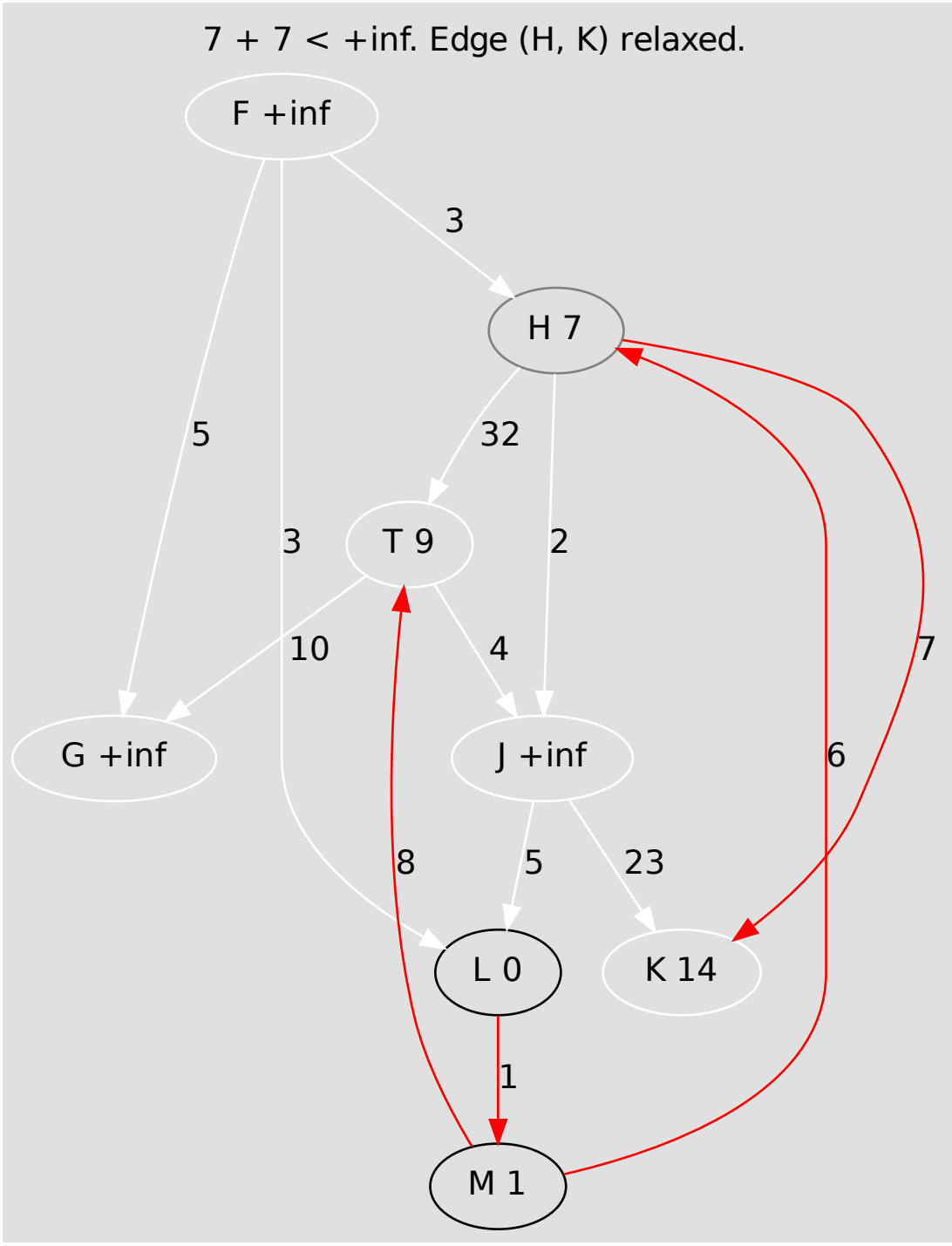

Relaxing the edges from M. Considering edge (M, H), leading to H.
Priority queue = [T 9, H +inf, F +inf, J +inf, G +inf, K +inf]
F +inf
3
H +inf
5
32
3
T 9
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M 1

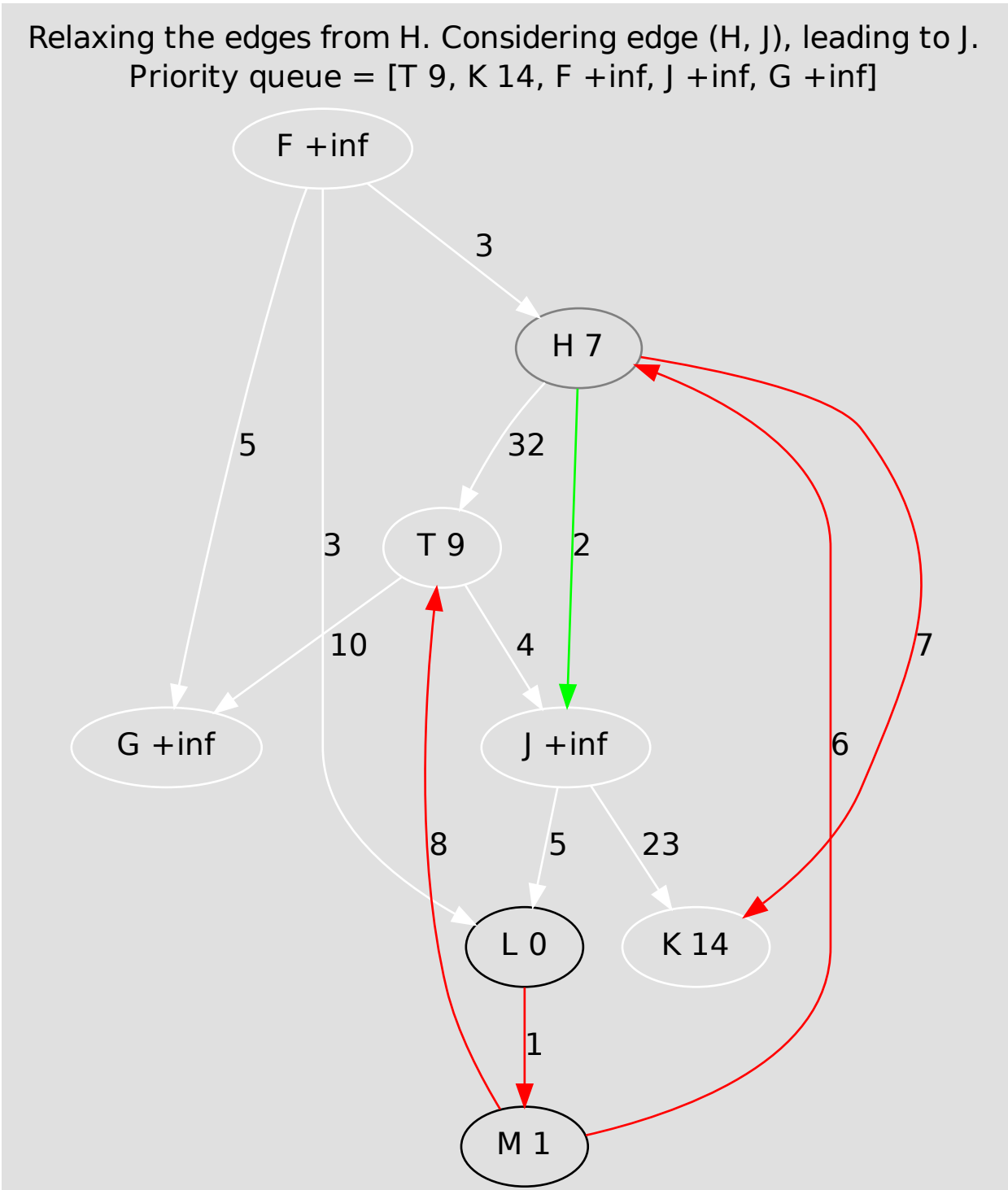

1 + 6 < +inf. Edge (M, H) relaxed.
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M 1

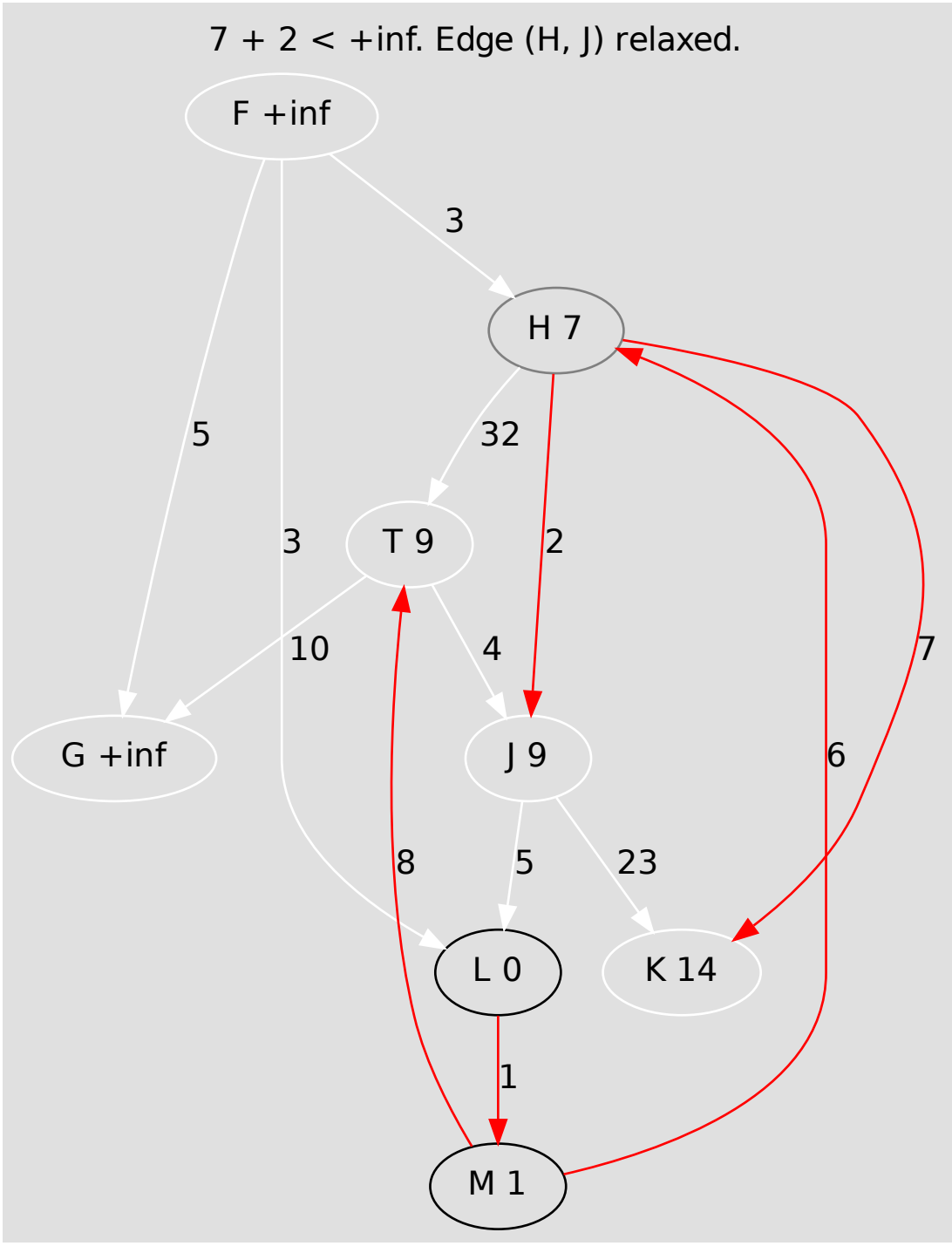

Finished with M.
Priority queue = [H 7, T 9, F +inf, J +inf, G +inf, K +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M 1

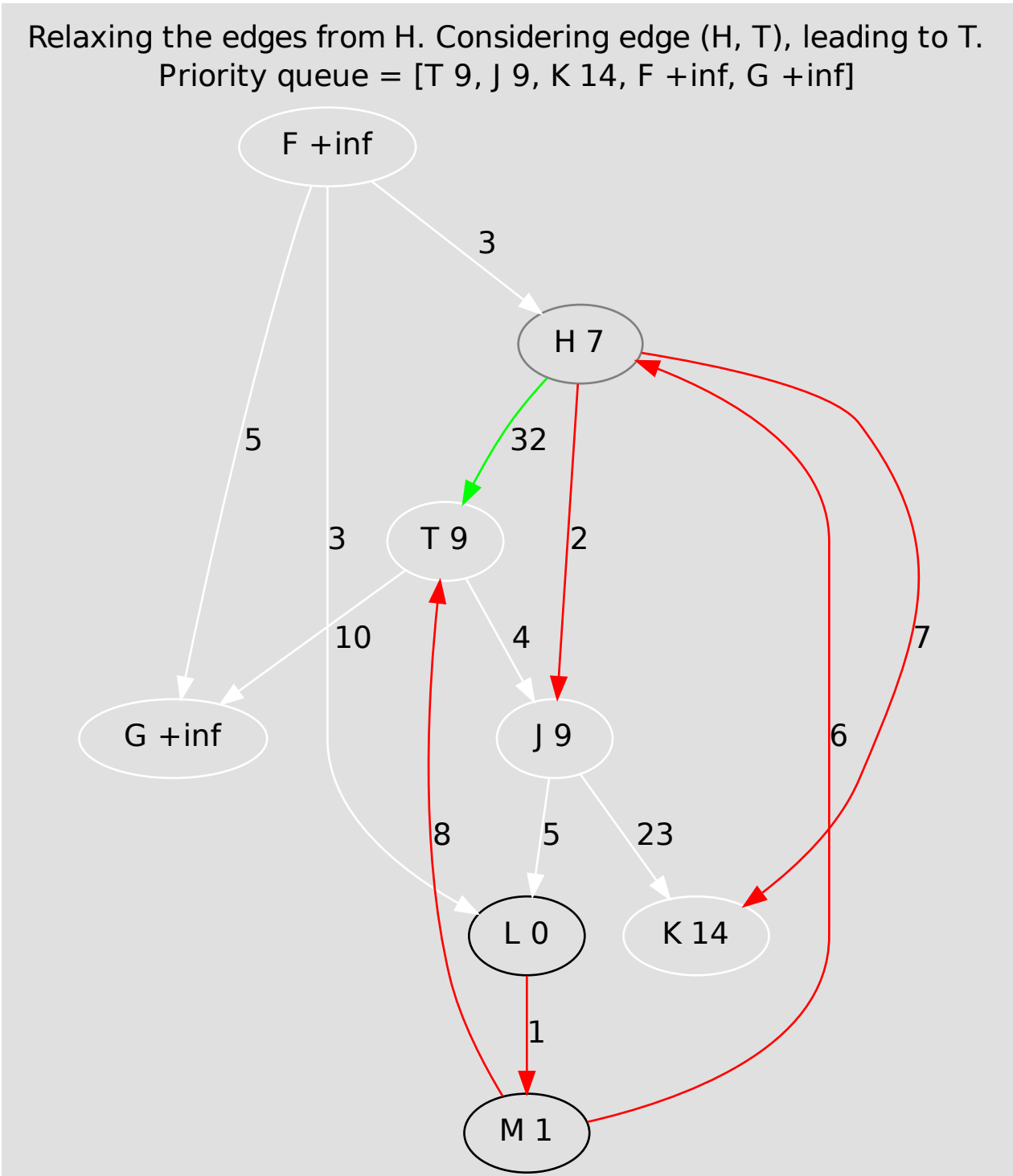

extractMin() -> H.
Priority queue = [T 9, F +inf, J +inf, G +inf, K +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M 1

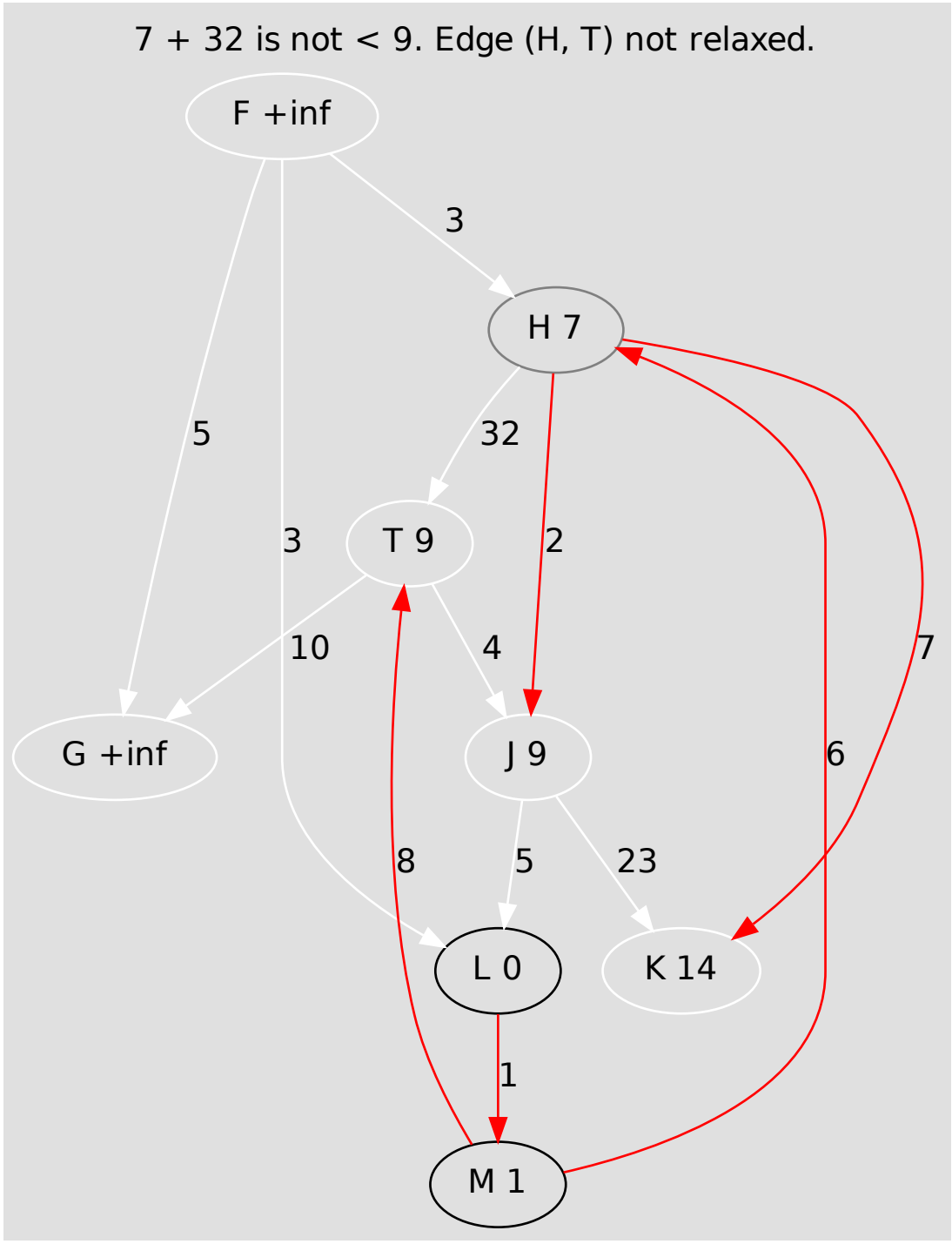

Relaxing the edges from H. Considering edge (H, K), leading to K.
Priority queue = [T 9, F +inf, J +inf, G +inf, K +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K +inf
1
M 1

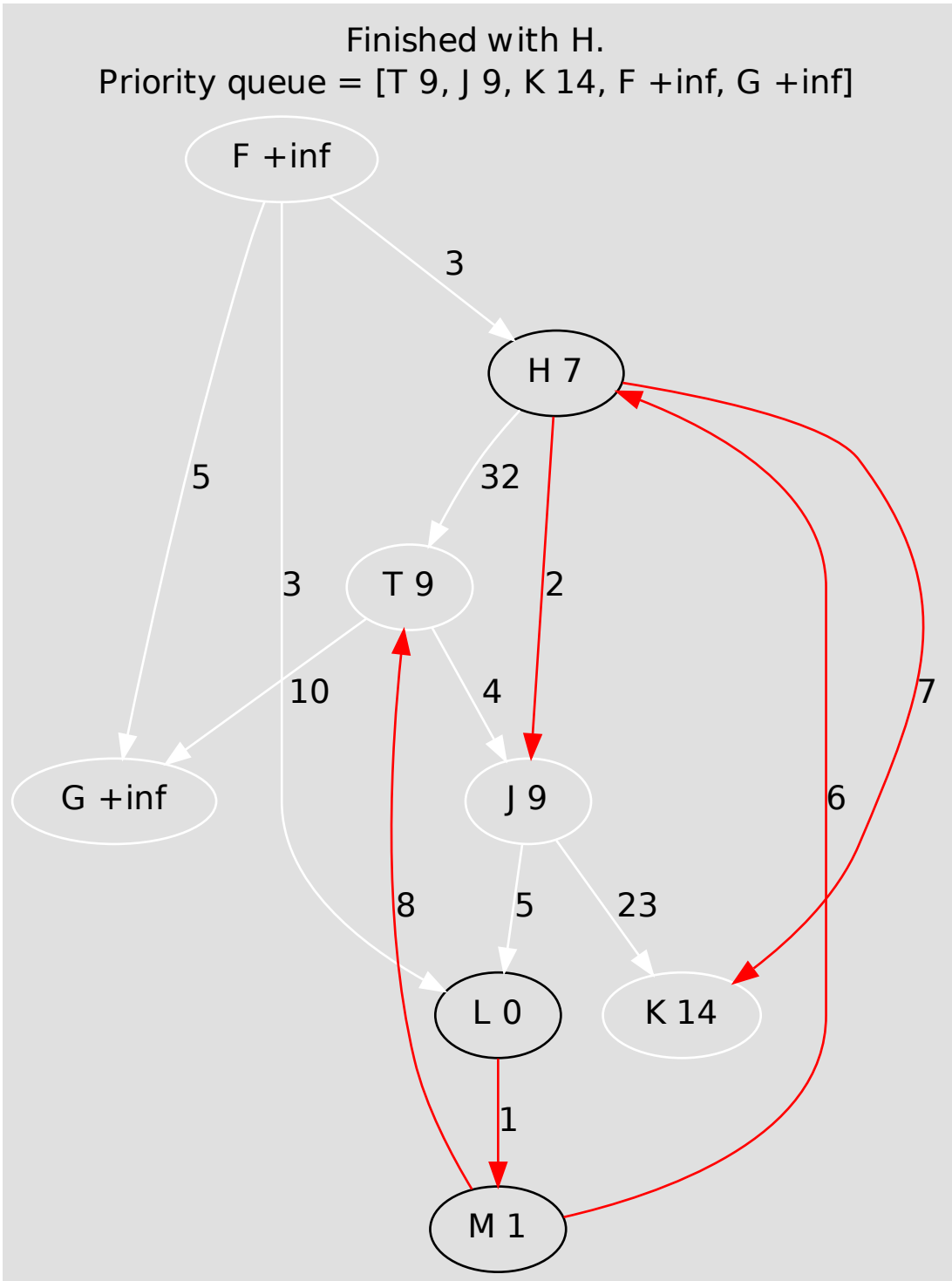

7 + 7 < +inf. Edge (H, K) relaxed.
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K 14
1
M 1

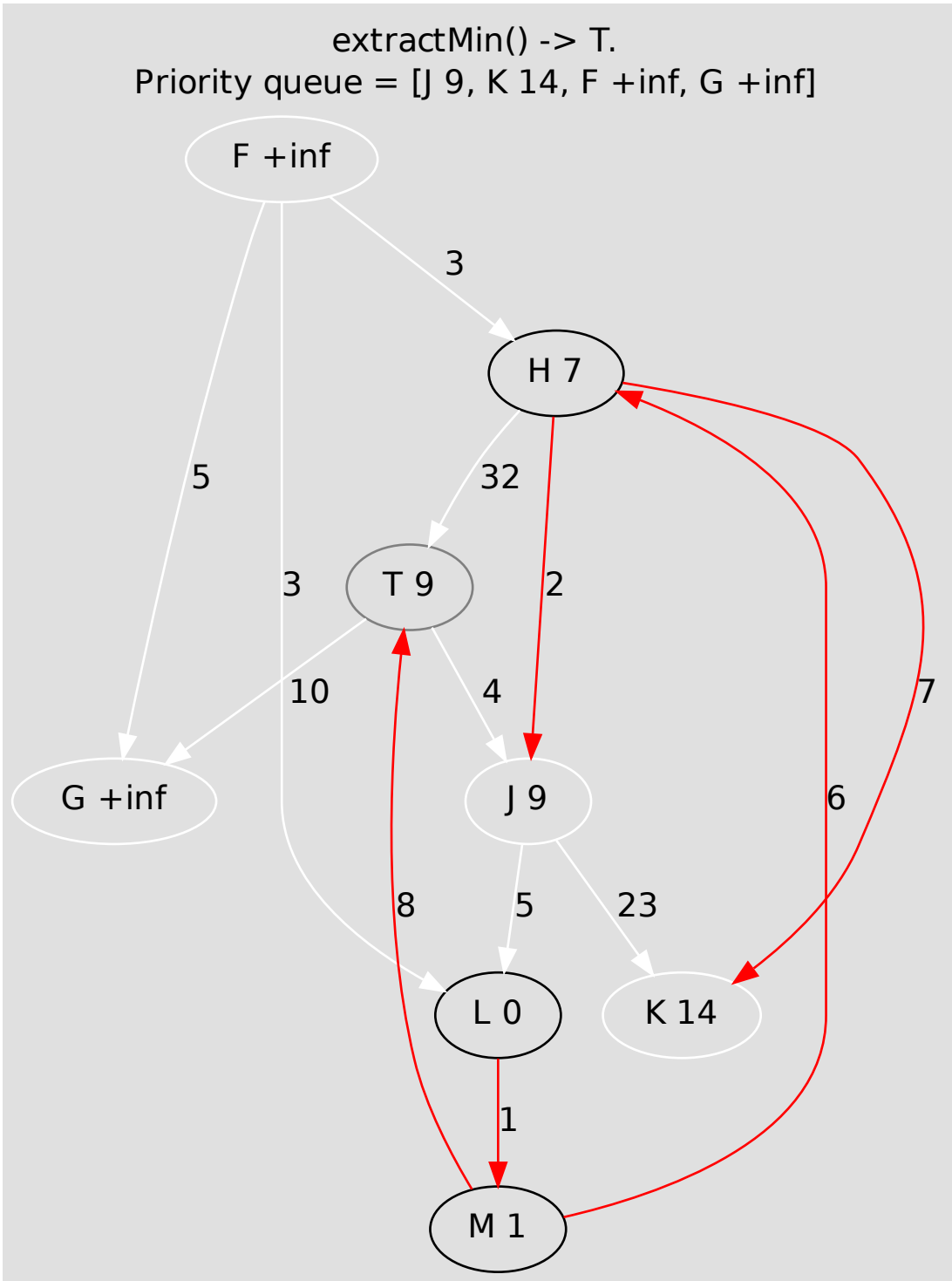

Relaxing the edges from H. Considering edge (H, J), leading to J.
Priority queue = [T 9, K 14, F +inf, J +inf, G +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J +inf
6
8
5
23
L 0
K 14
1
M 1

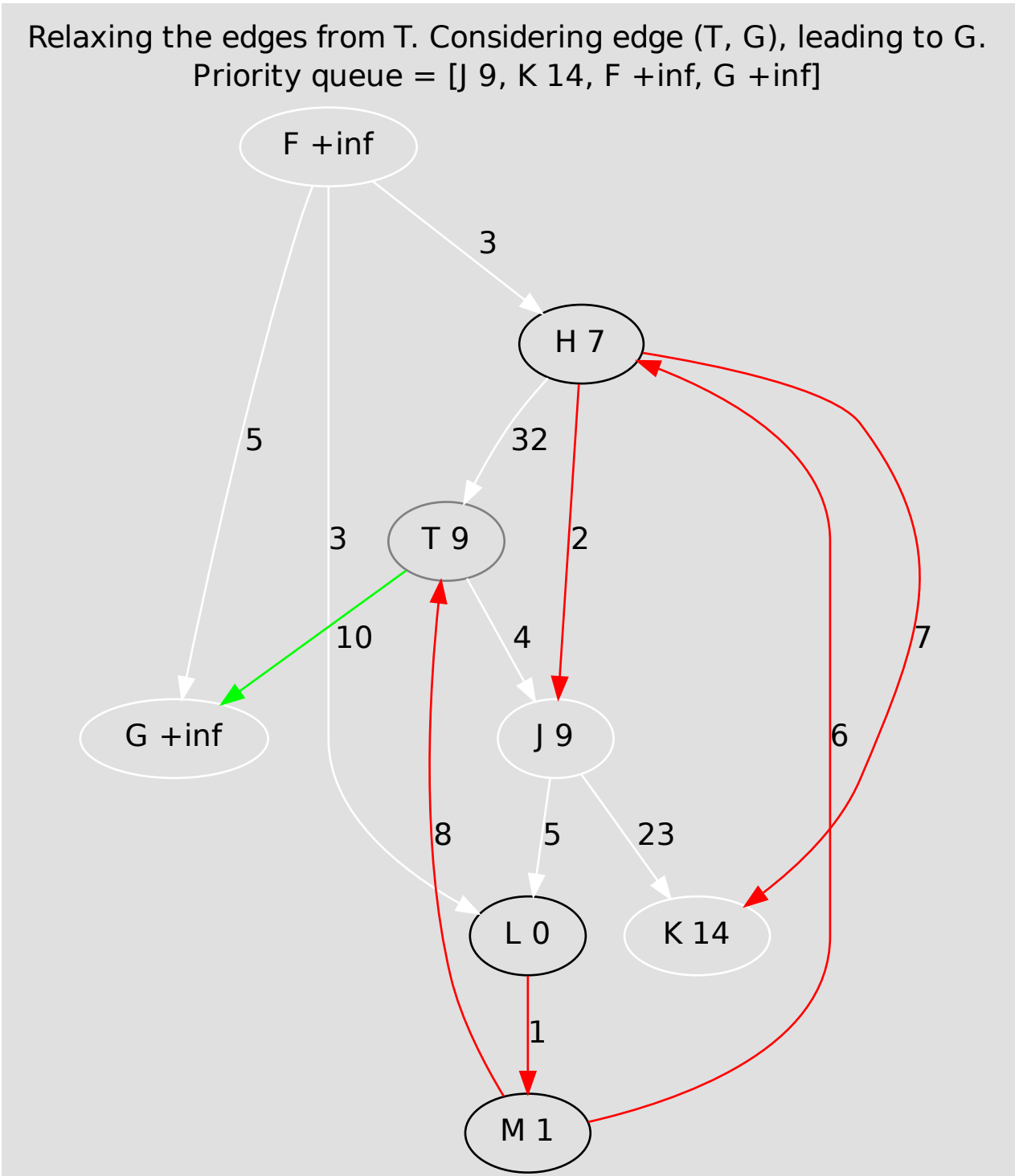

7 + 2 < +inf. Edge (H, J) relaxed.
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J 9
6
8
5
23
L 0
K 14
1
M 1

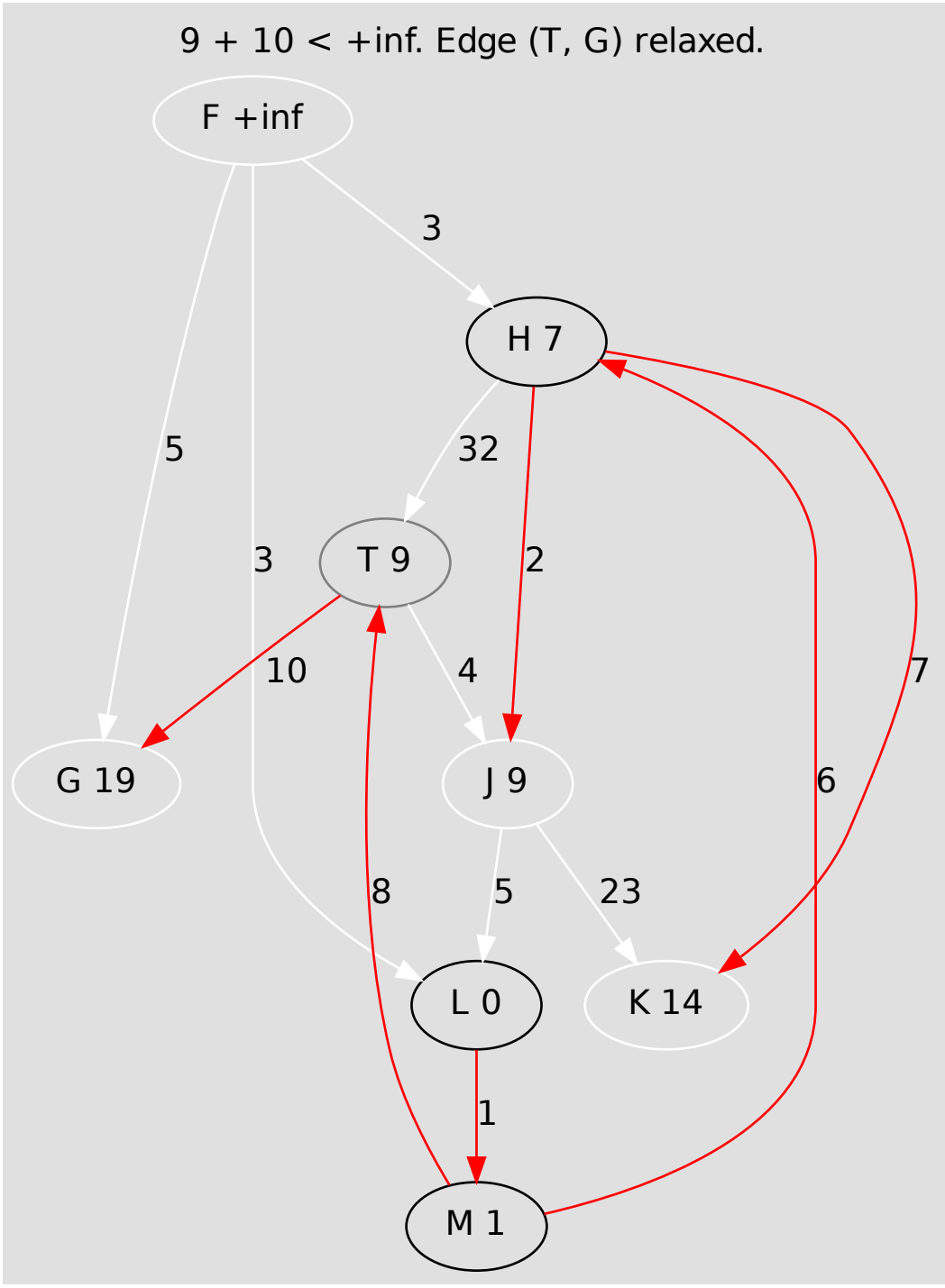

Relaxing the edges from H. Considering edge (H, T), leading to T.
Priority queue = [T 9, J 9, K 14, F +inf, G +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J 9
6
8
5
23
L 0
K 14
1
M 1

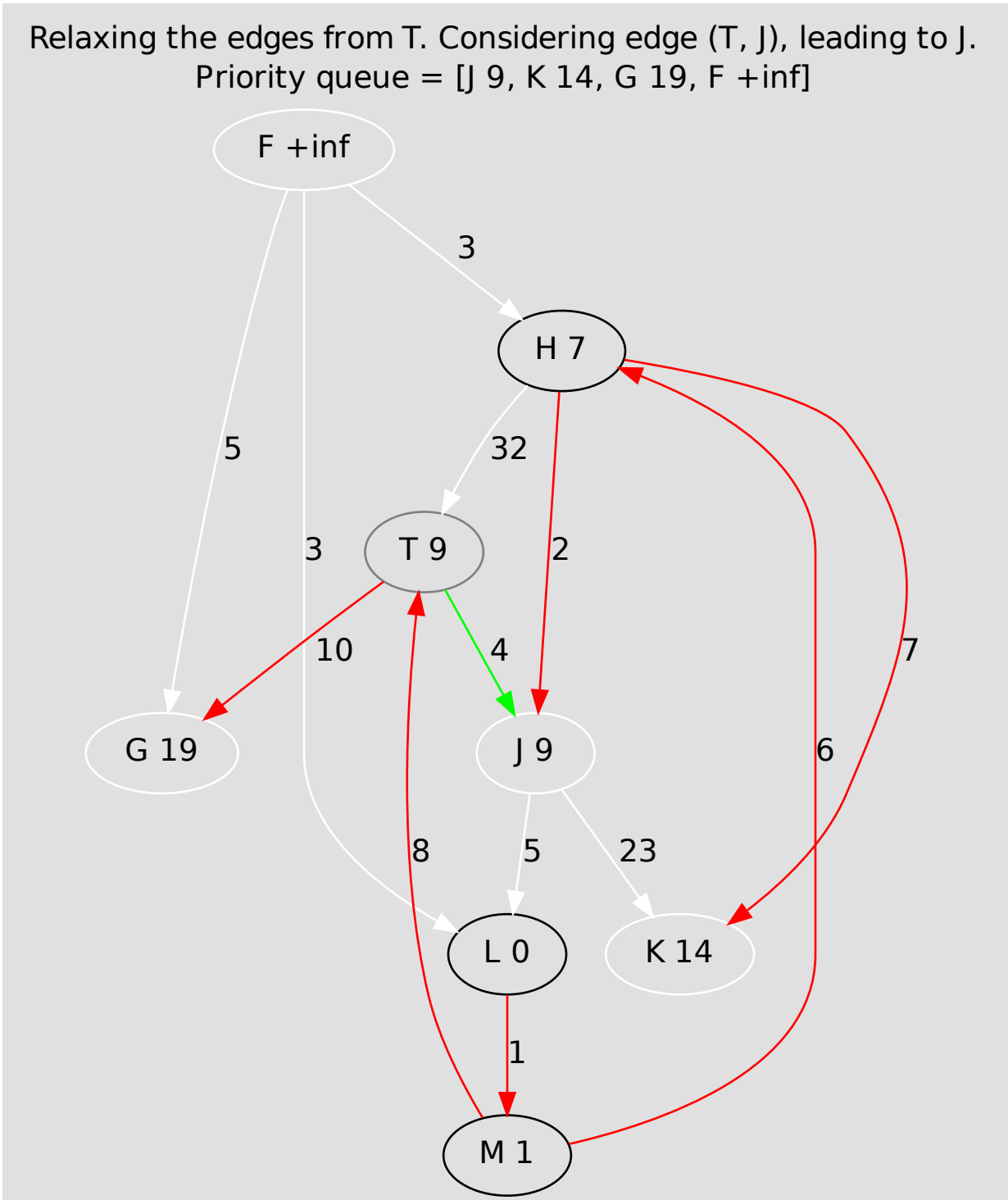

7 + 32 is not < 9. Edge (H, T) not relaxed.
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J 9
6
8
5
23
L 0
K 14
1
M 1

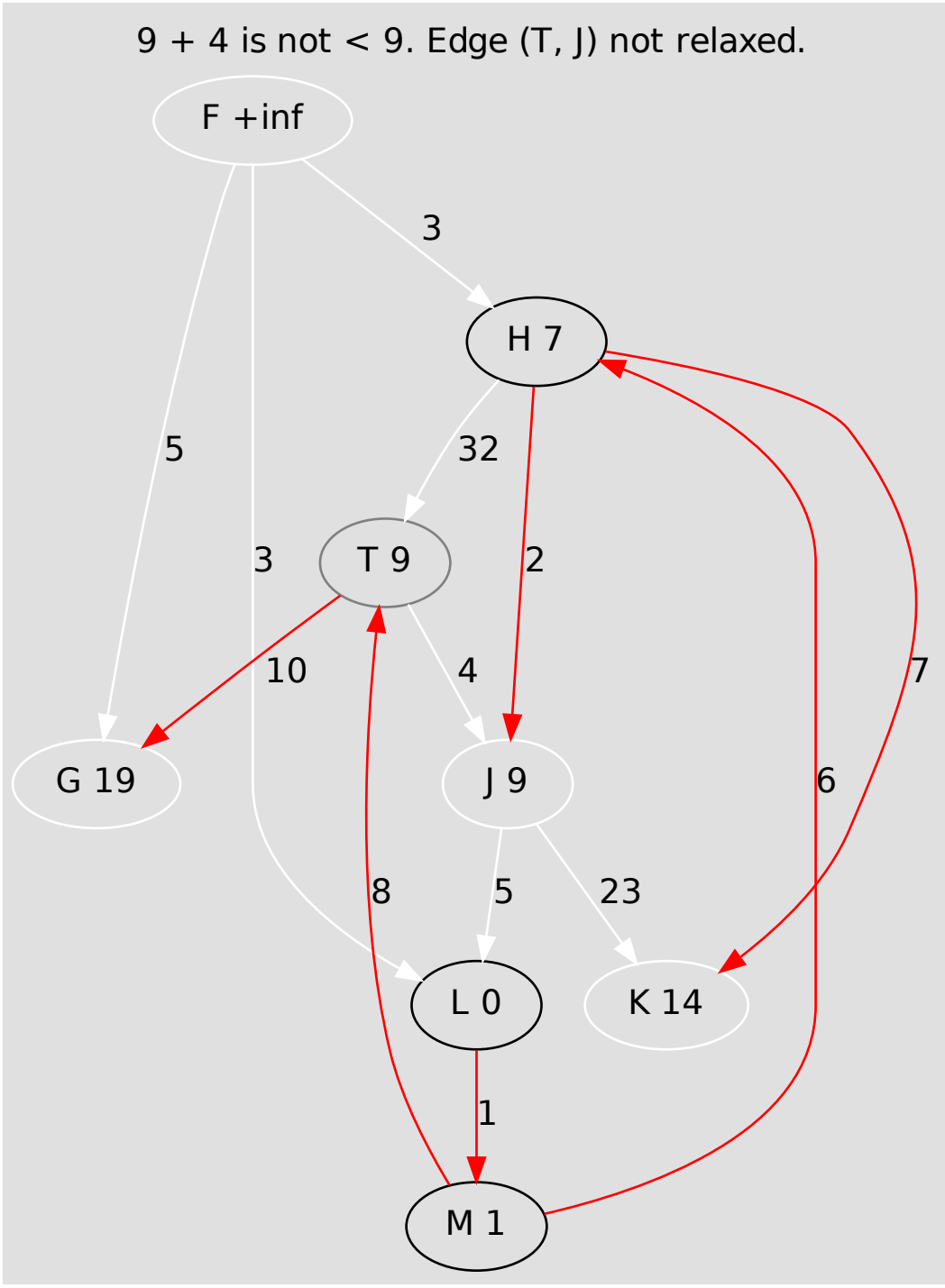

Finished with H.
Priority queue = [T 9, J 9, K 14, F +inf, G +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J 9
6
8
5
23
L 0
K 14
1
M 1

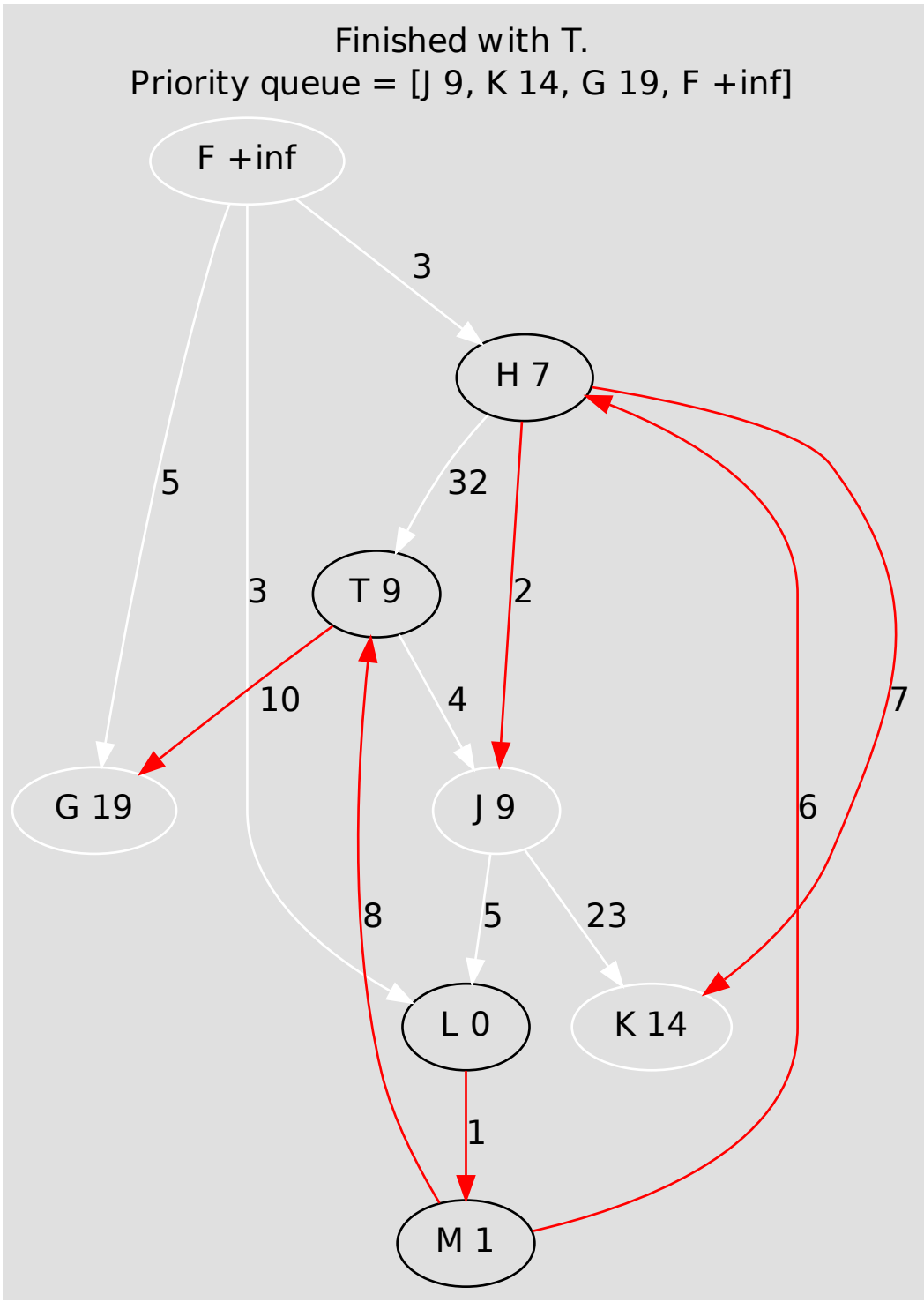

extractMin() -> T.
Priority queue = [J 9, K 14, F +inf, G +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J 9
6
8
5
23
L 0
K 14
1
M 1

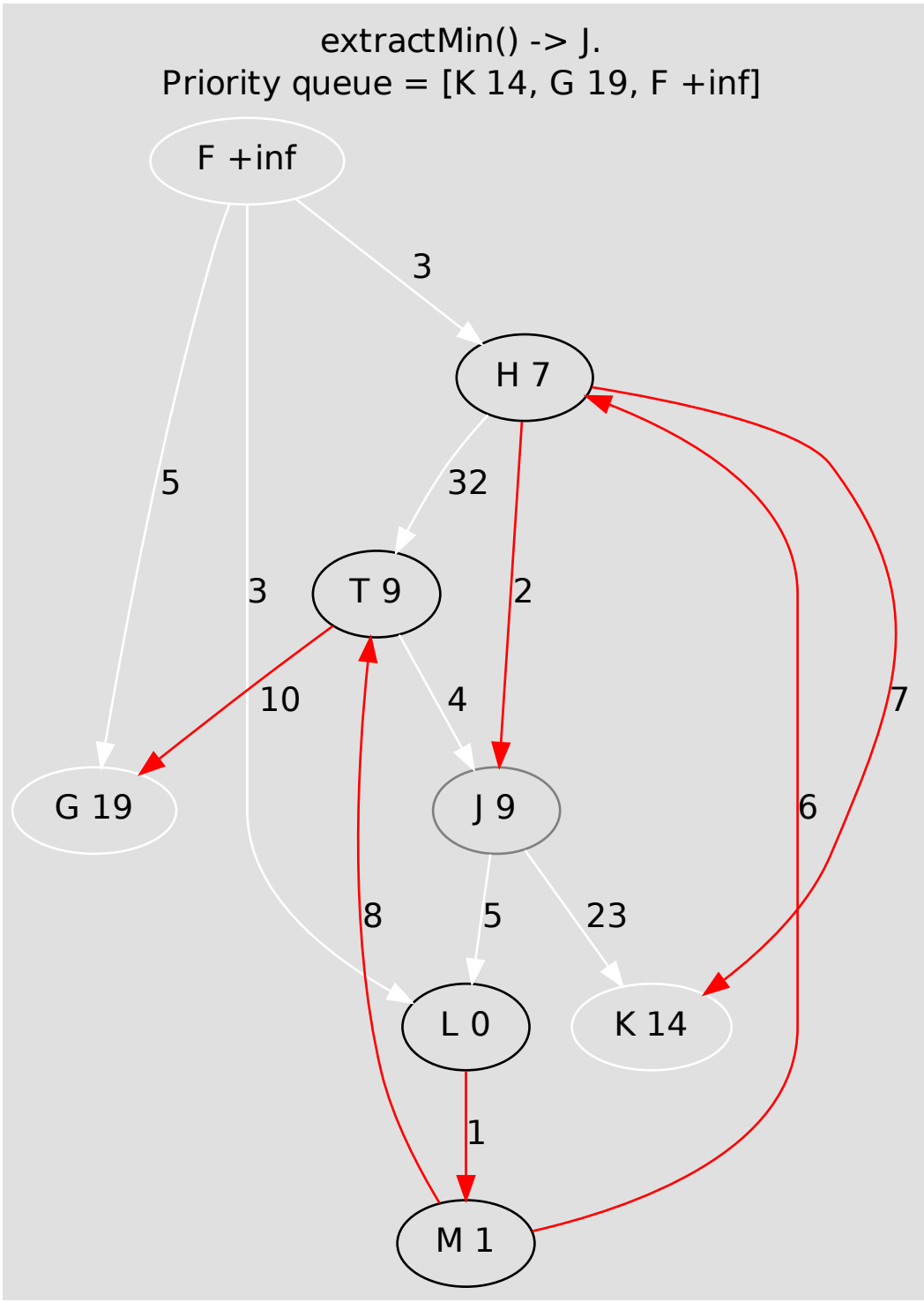

Relaxing the edges from T. Considering edge (T, G), leading to G.
Priority queue = [J 9, K 14, F +inf, G +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G +inf
J 9
6
8
5
23
L 0
K 14
1
M 1


9 + 10 < +inf. Edge (T, G) relaxed.
F +inf
H 7
T 9
G 19
J 9
L 0
K 14
M 1
3
5
32
3
2
10
4
7
6
8
5
23
1

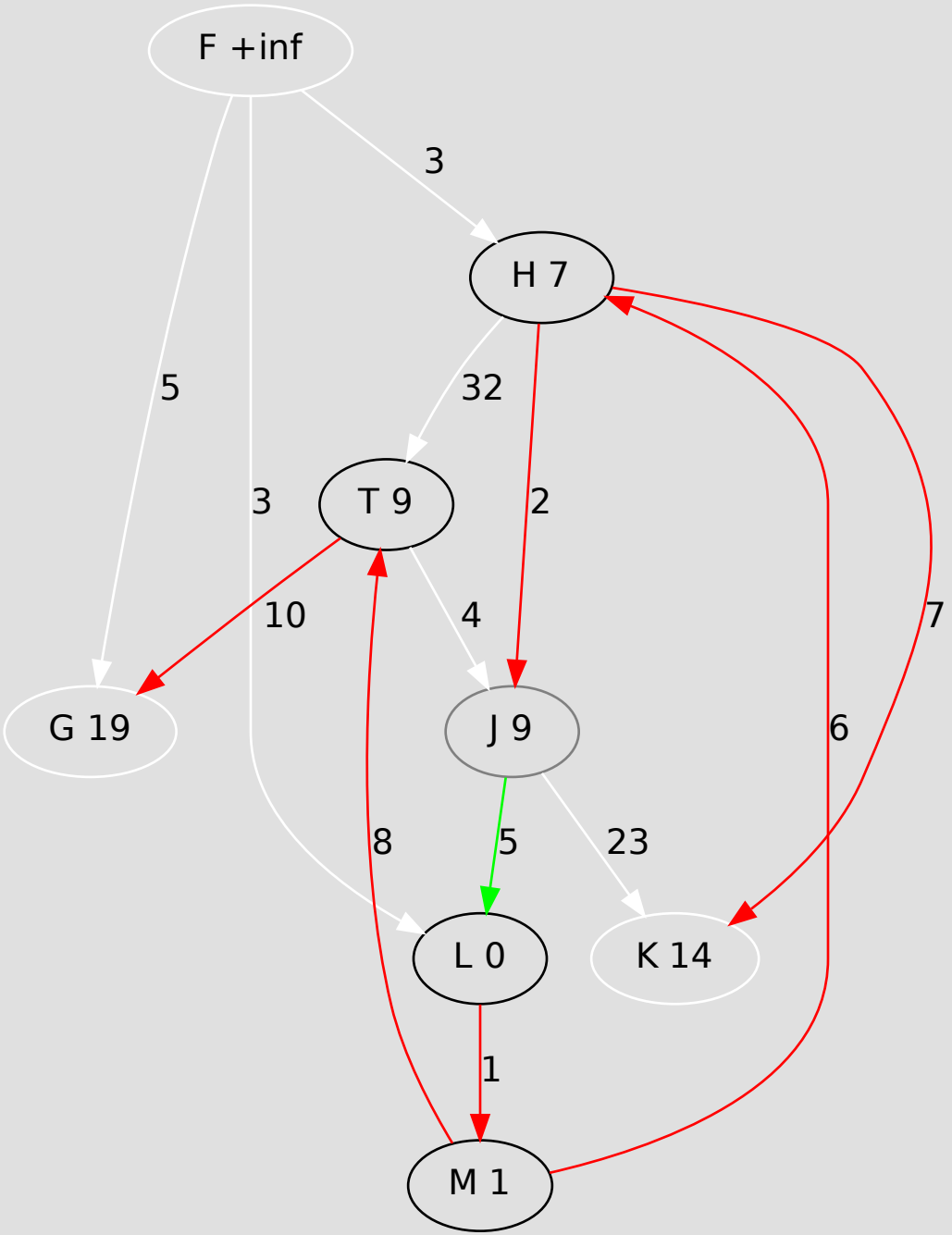

Relaxing the edges from T. Considering edge (T, J), leading to J.
Priority queue = [J 9, K 14, G 19, F +inf]
F +inf
H 7
T 9
G 19
J 9
L 0
K 14
M 1
3
5
32
3
2
10
4
7
6
8
5
23
1

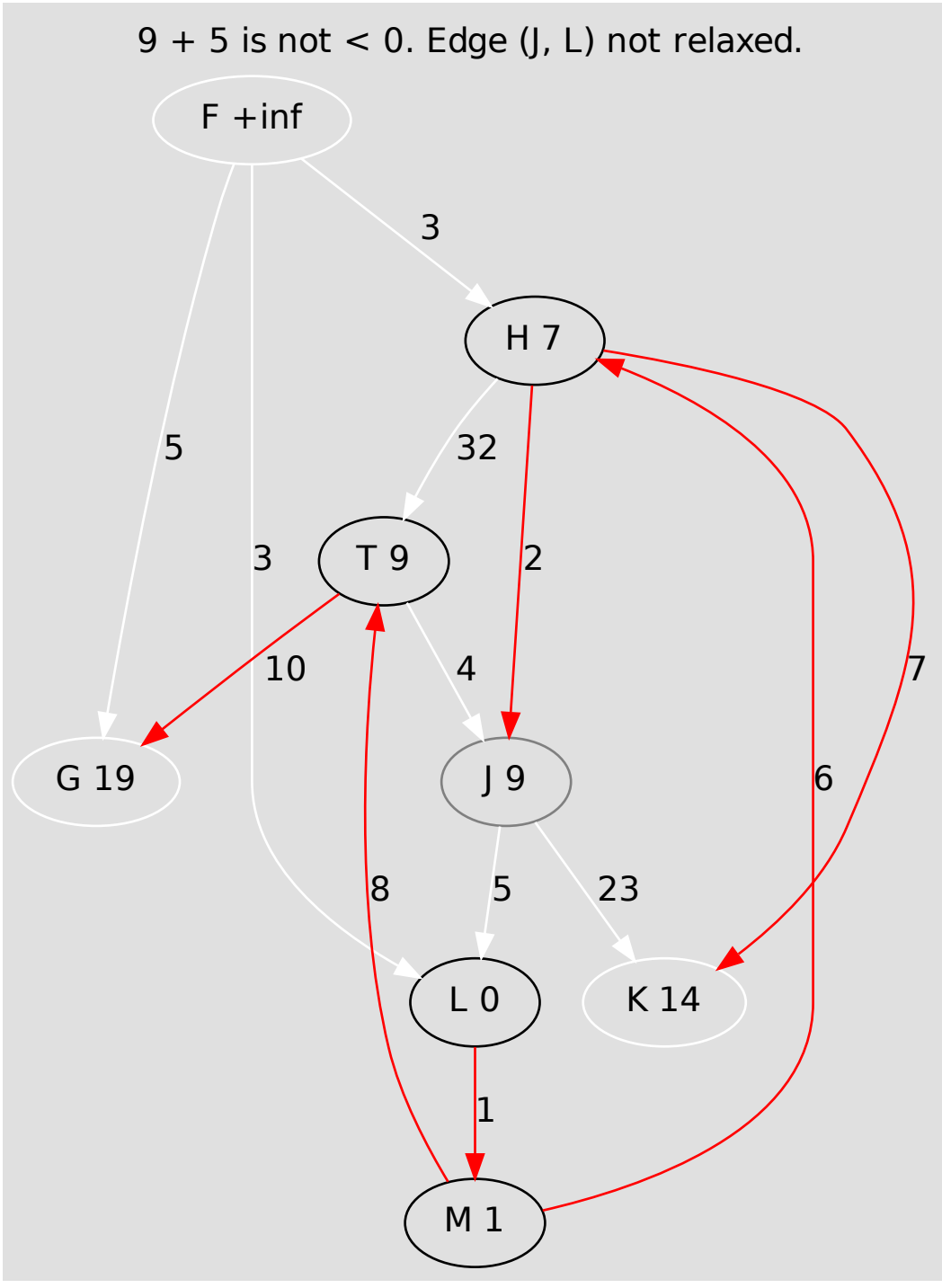

9 + 4 is not < 9. Edge (T, J) not relaxed.
F +inf
3
H 7
5
32
T 9
3
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1

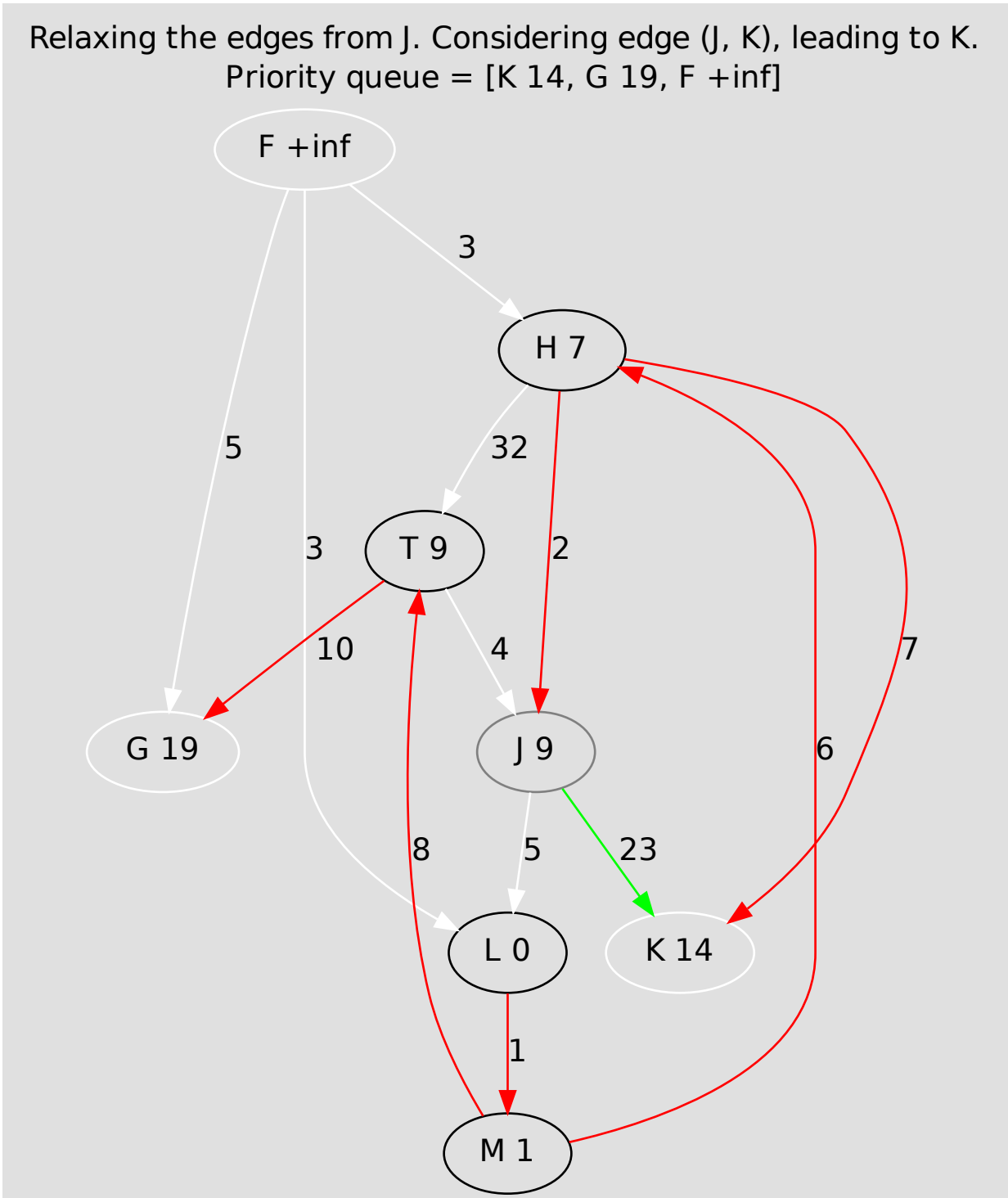

Finished with T.
Priority queue = [J 9, K 14, G 19, F +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1

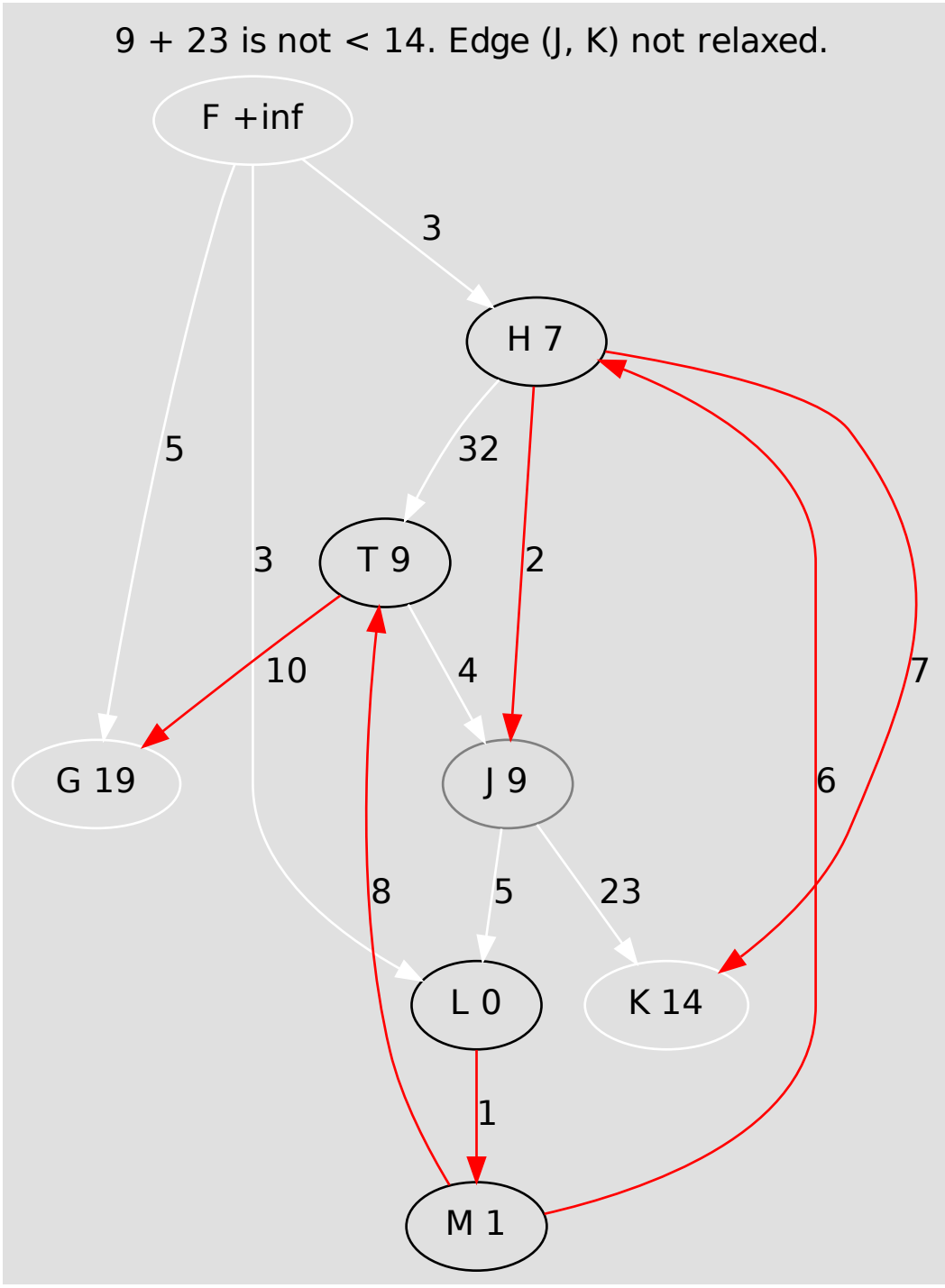

extractMin() -> J.
Priority queue = [K 14, G 19, F +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1

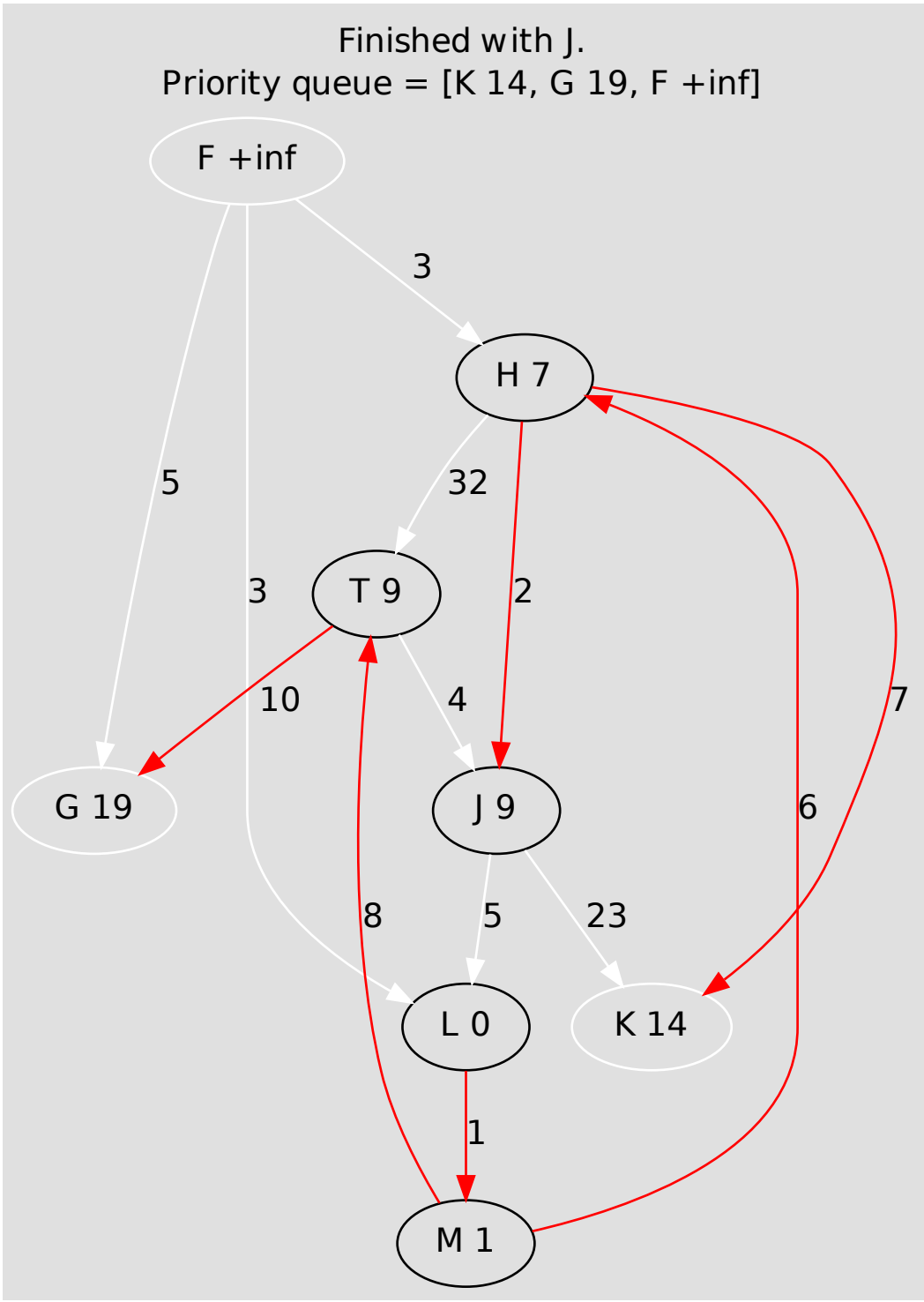
Relaxing the edges from J. Considering edge (J, L), leading to L.
Priority queue = [K 14, G 19, F +inf]

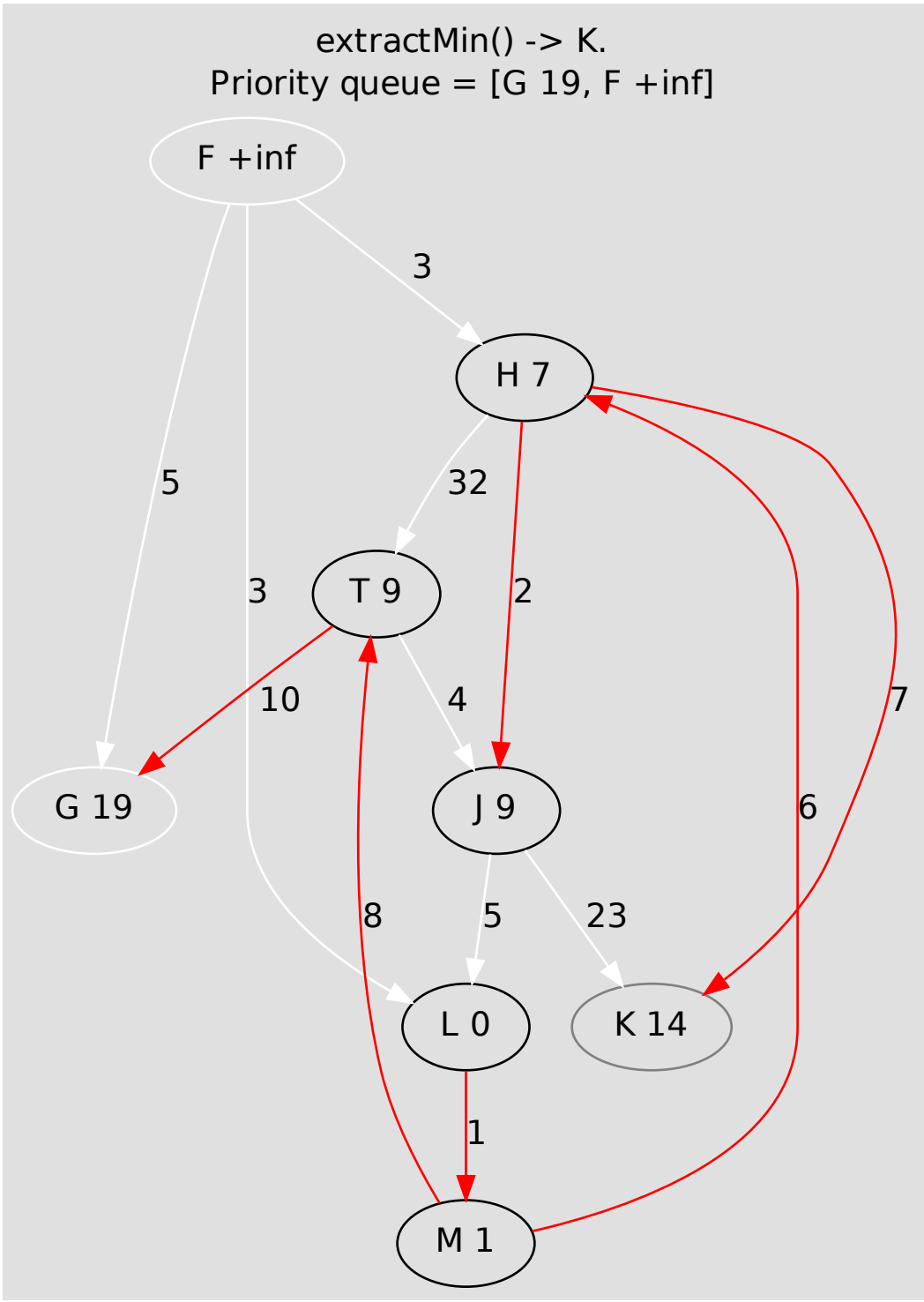


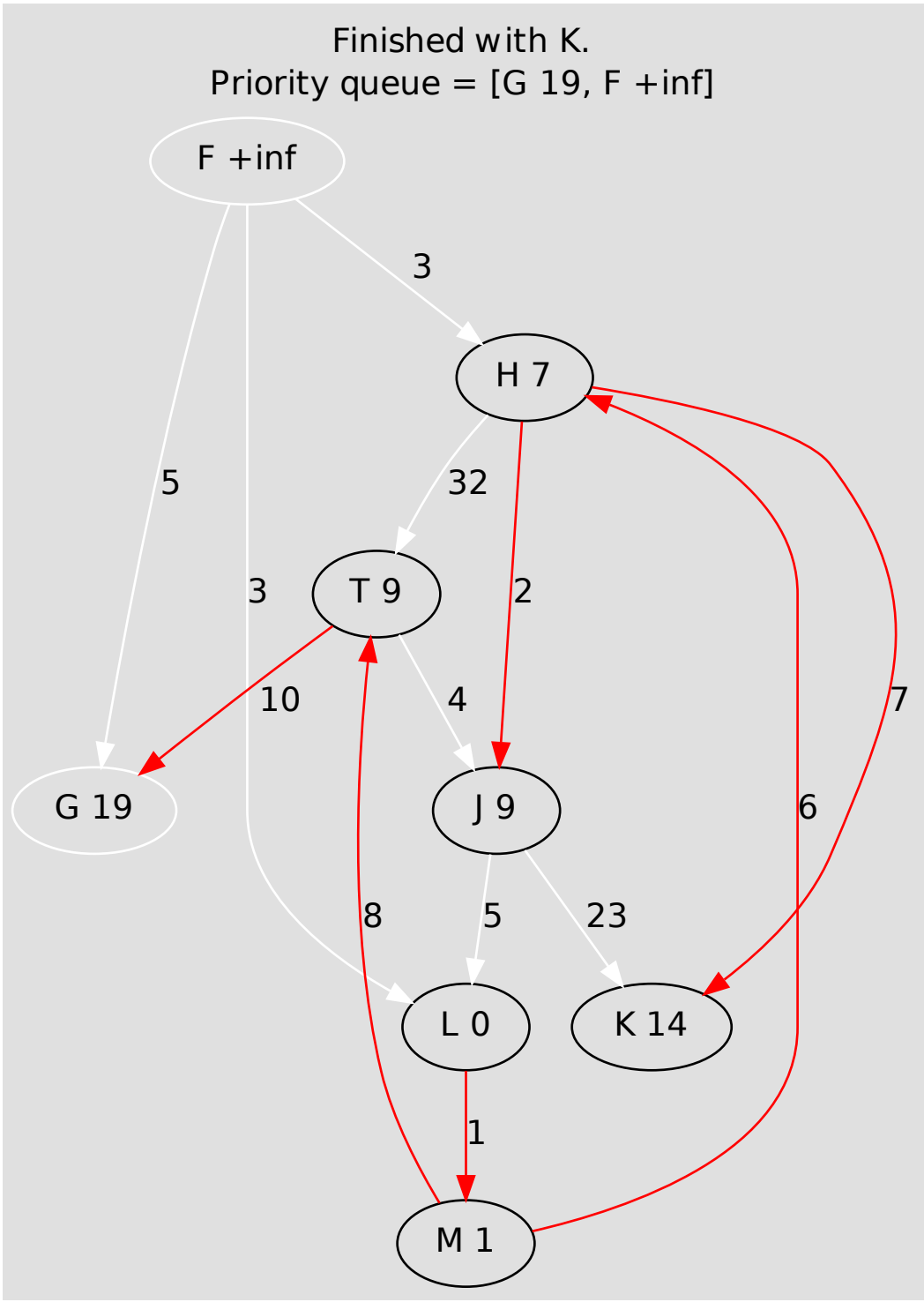

9 + 5 is not < 0. Edge (J, L) not relaxed.
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1

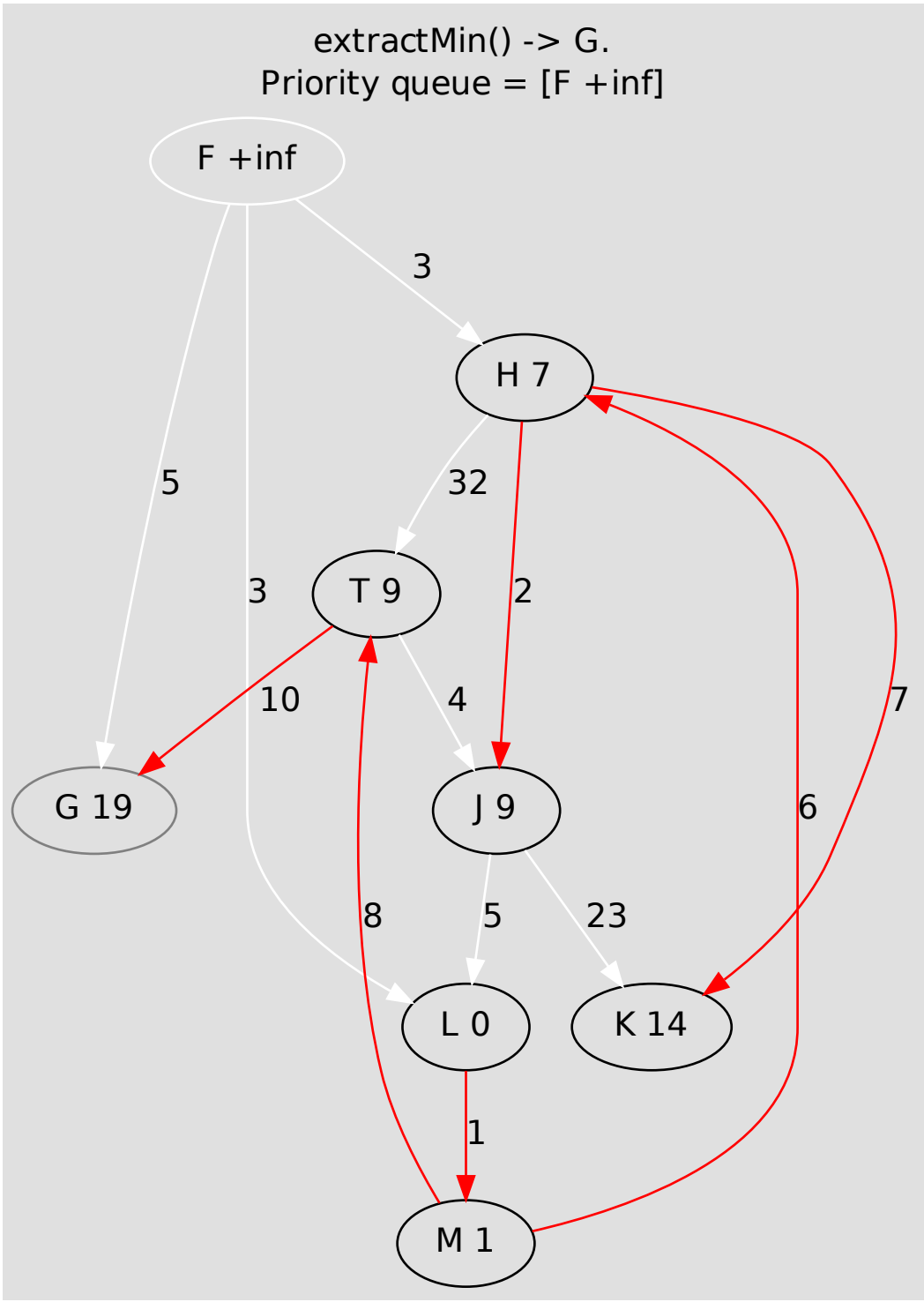

Relaxing the edges from J. Considering edge (J, K), leading to K.
Priority queue = [K 14, G 19, F +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1

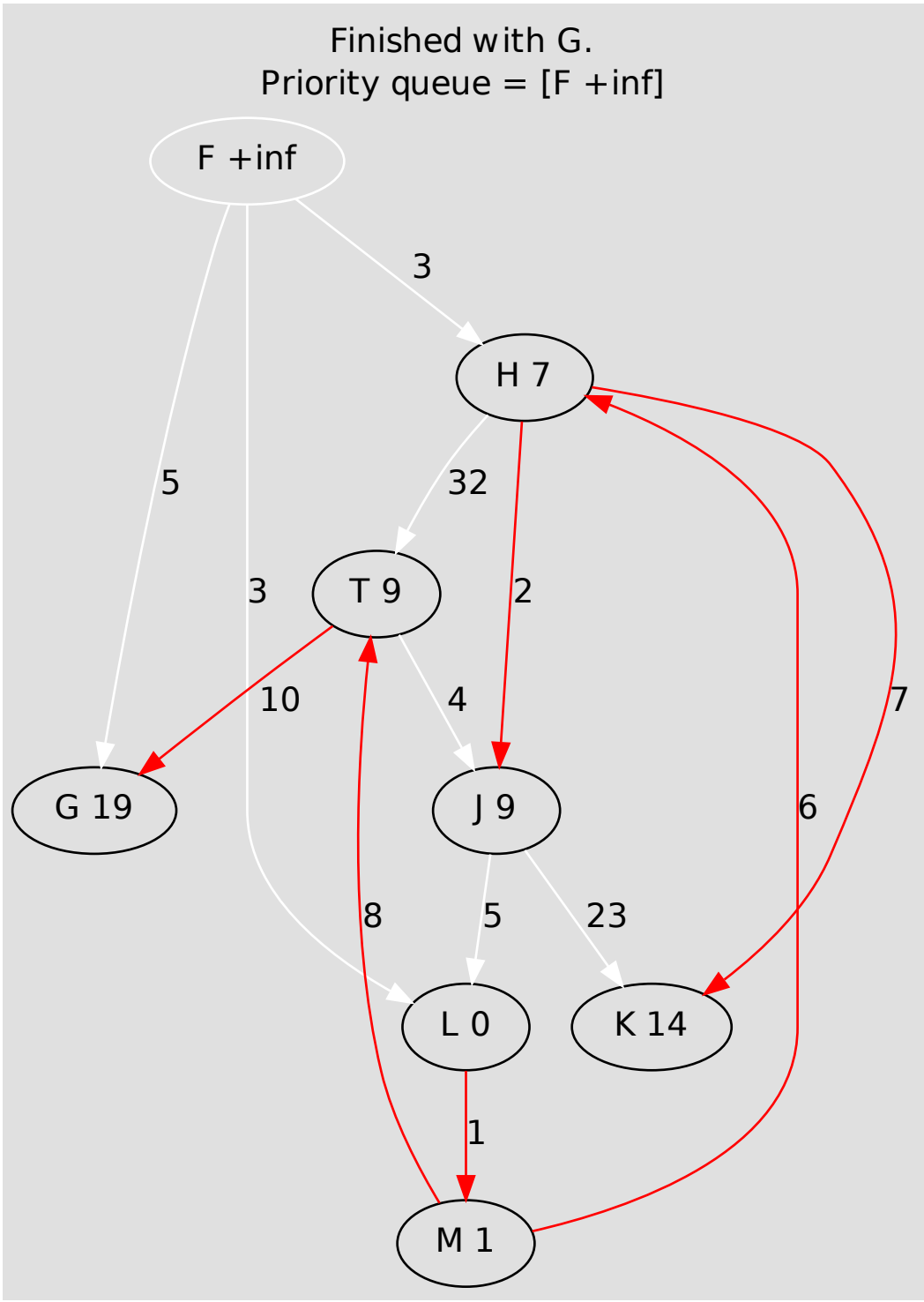

9 + 23 is not < 14. Edge (J, K) not relaxed.
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1

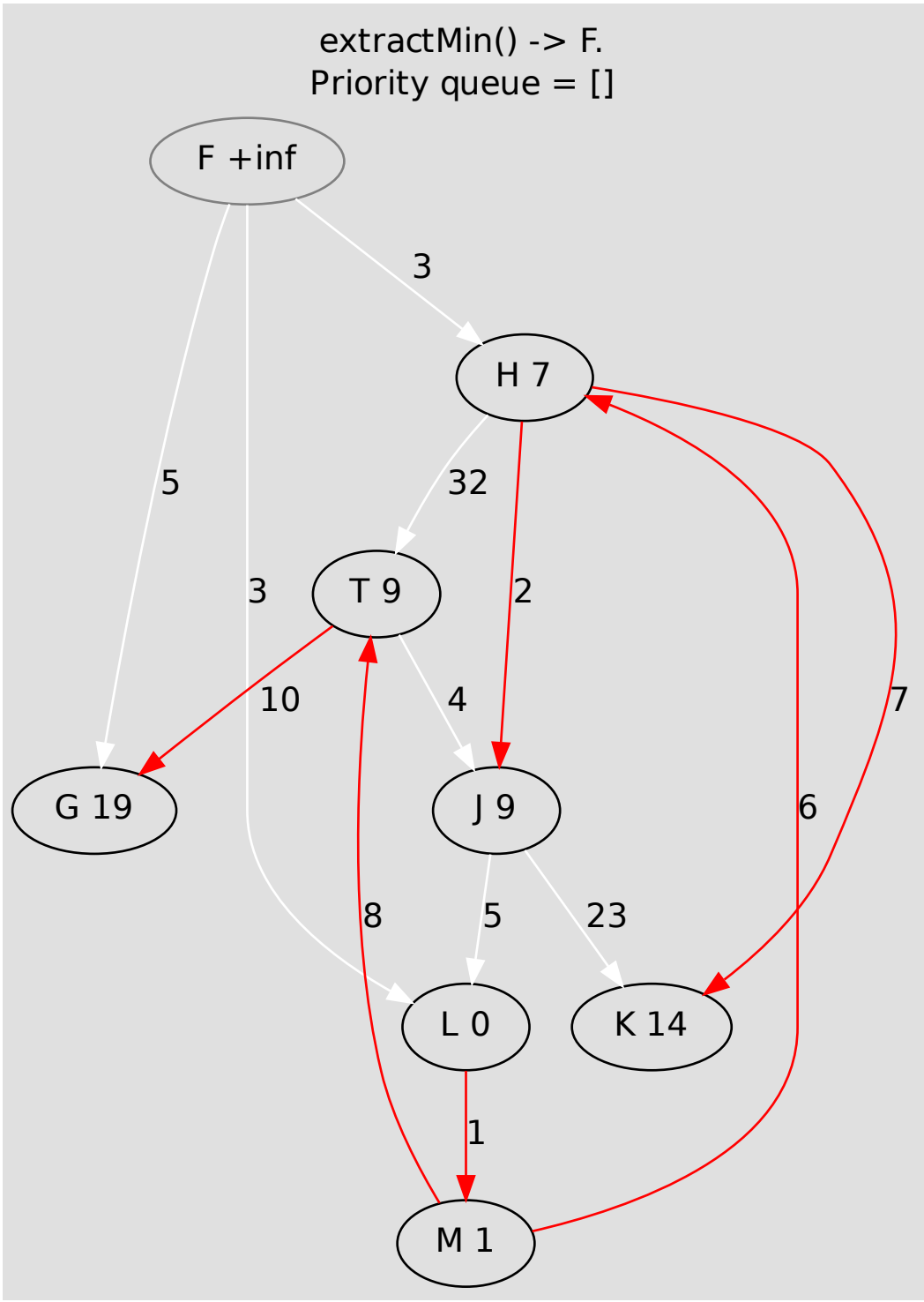

Finished with J.
Priority queue = [K 14, G 19, F +inf]
F +inf
H 7
T 9
G 19
J 9
L 0
K 14
M 1
3
5
32
3
2
10
4
7
6
8
5
23
1

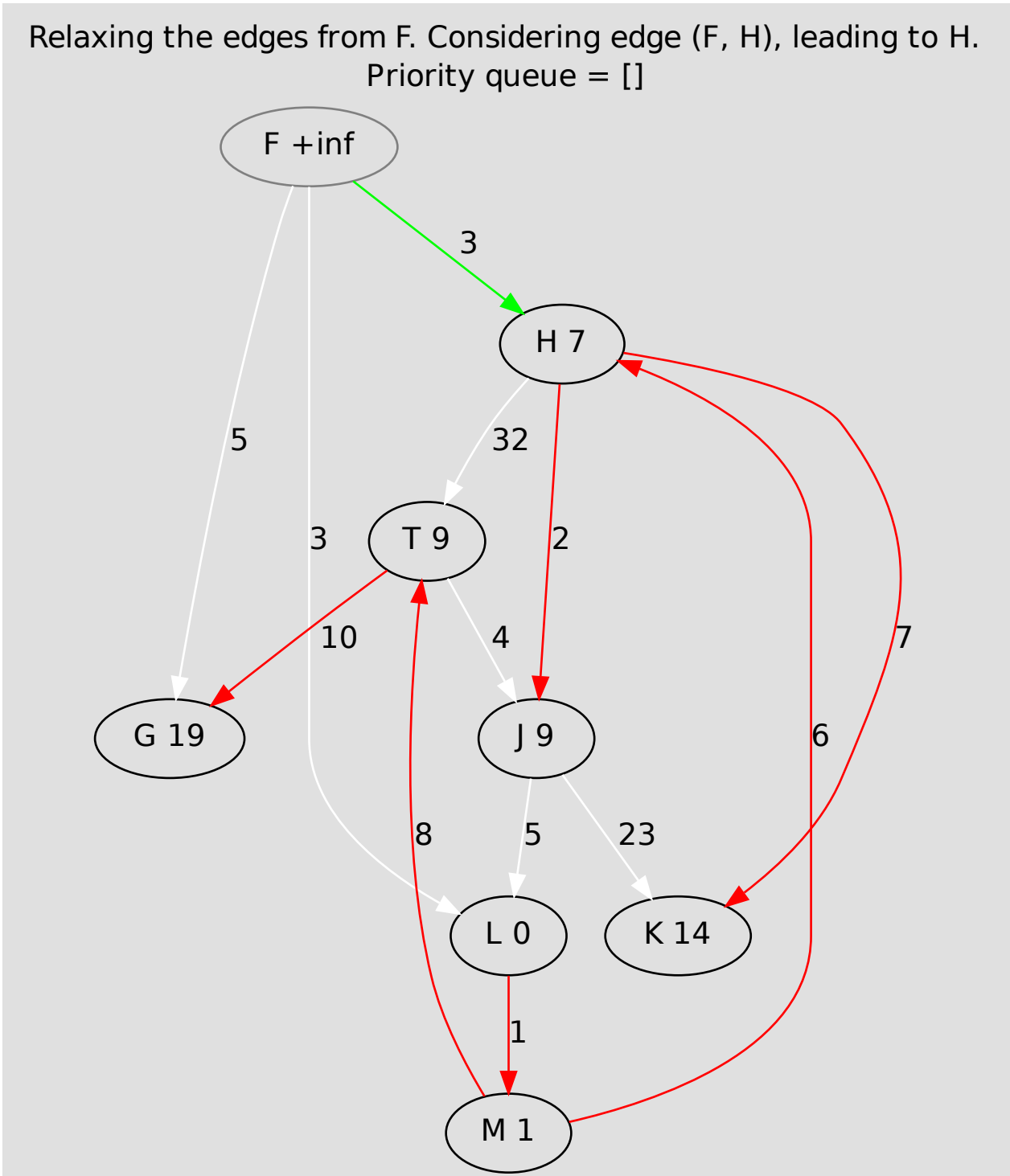

extractMin() -> K.
Priority queue = [G 19, F +inf]
F +inf
H 7
T 9
G 19
J 9
L 0
K 14
M 1
3
5
32
3
2
10
4
7
6
8
5
23
1

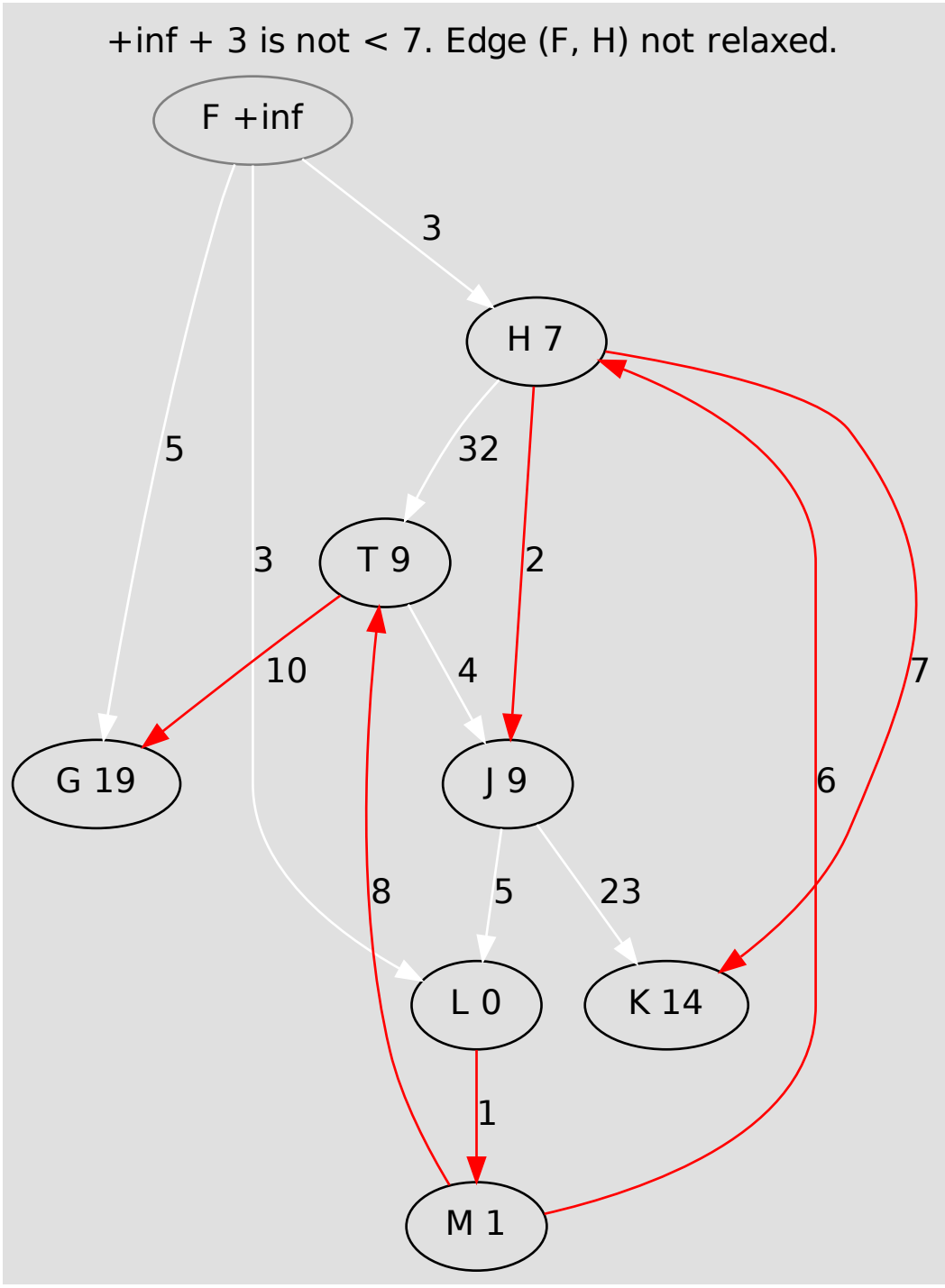

Finished with K.
Priority queue = [G 19, F +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1

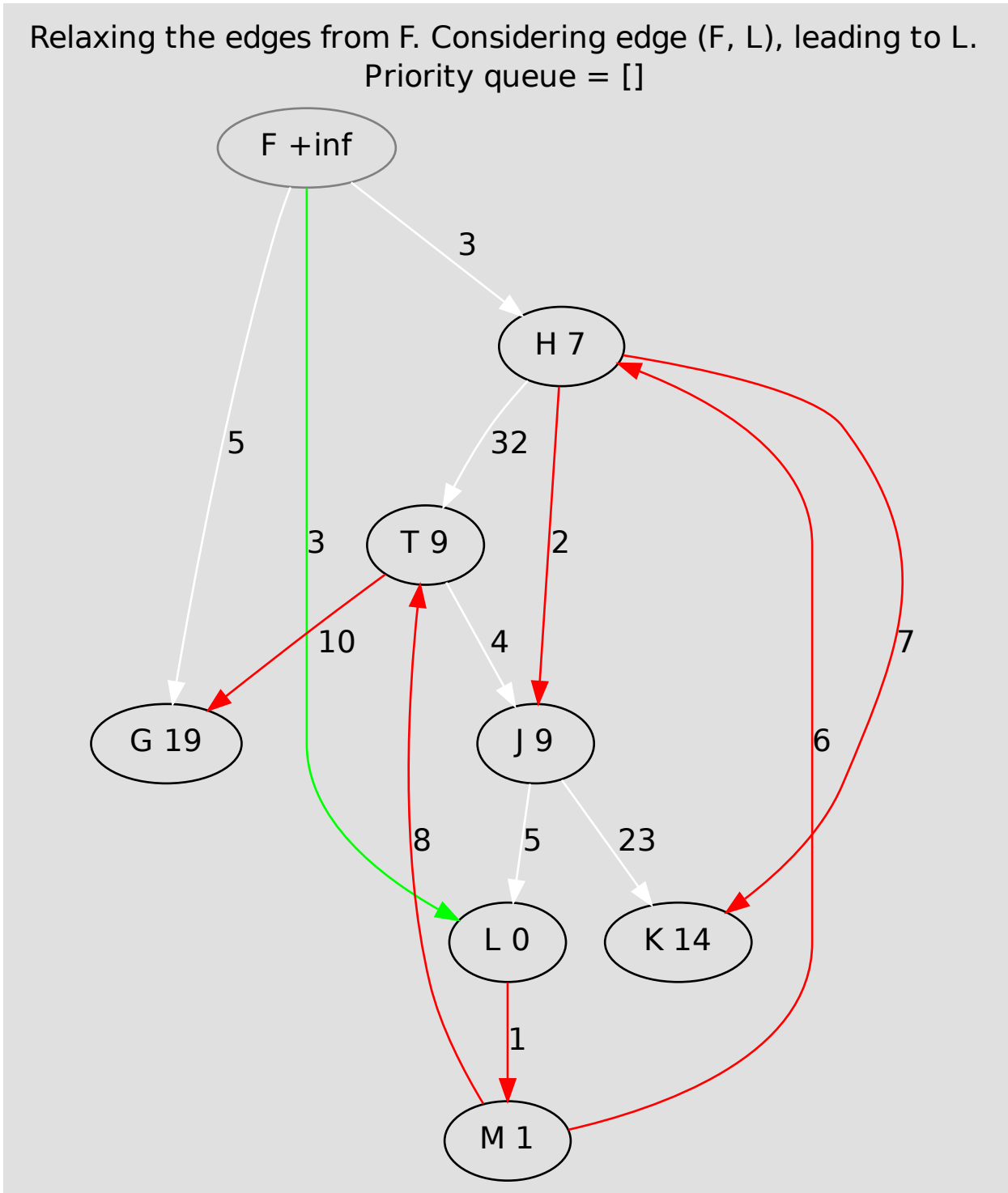

extractMin() -> G.
Priority queue = [F +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1

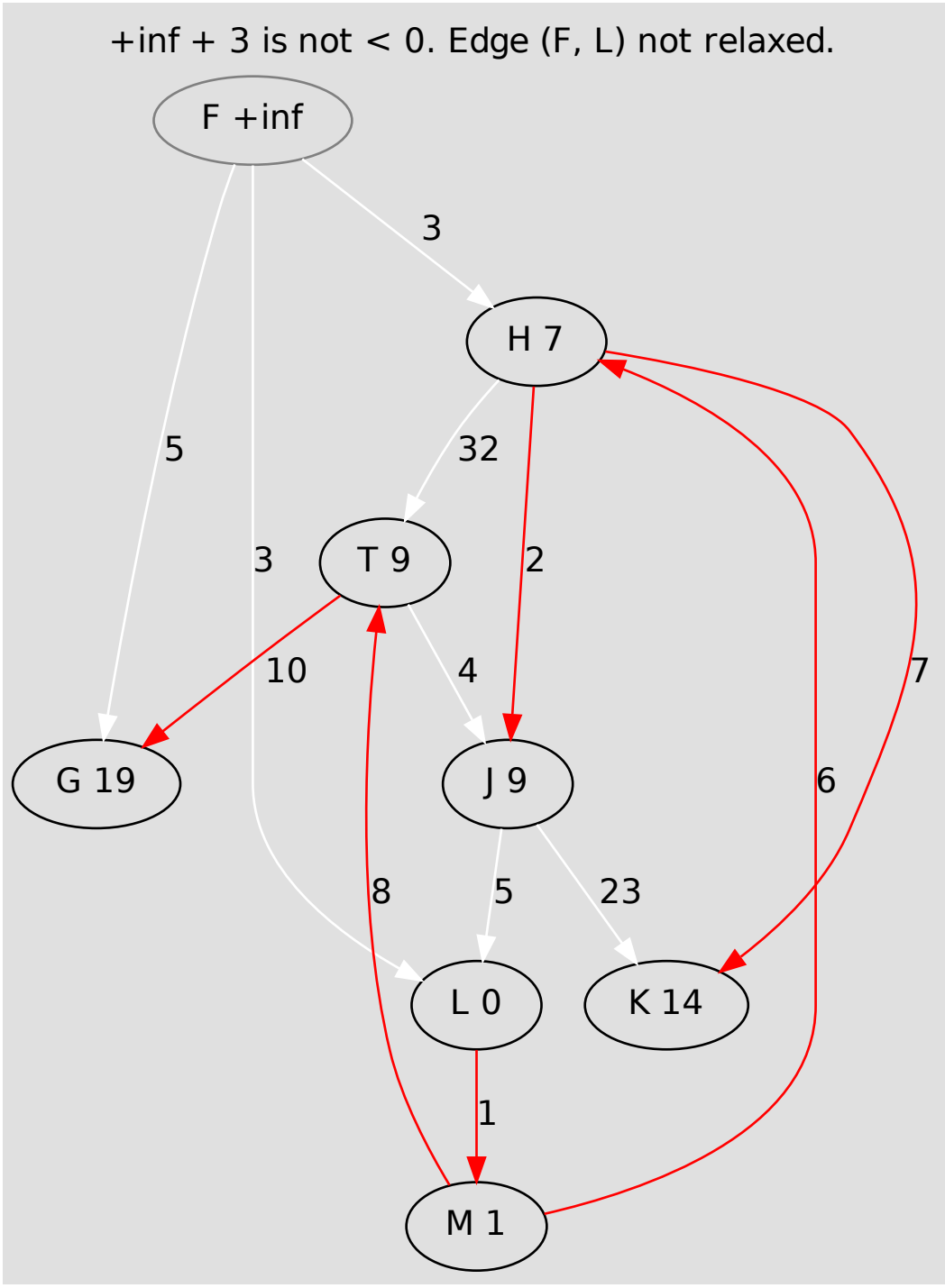

Finished with G.
Priority queue = [F +inf]
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1

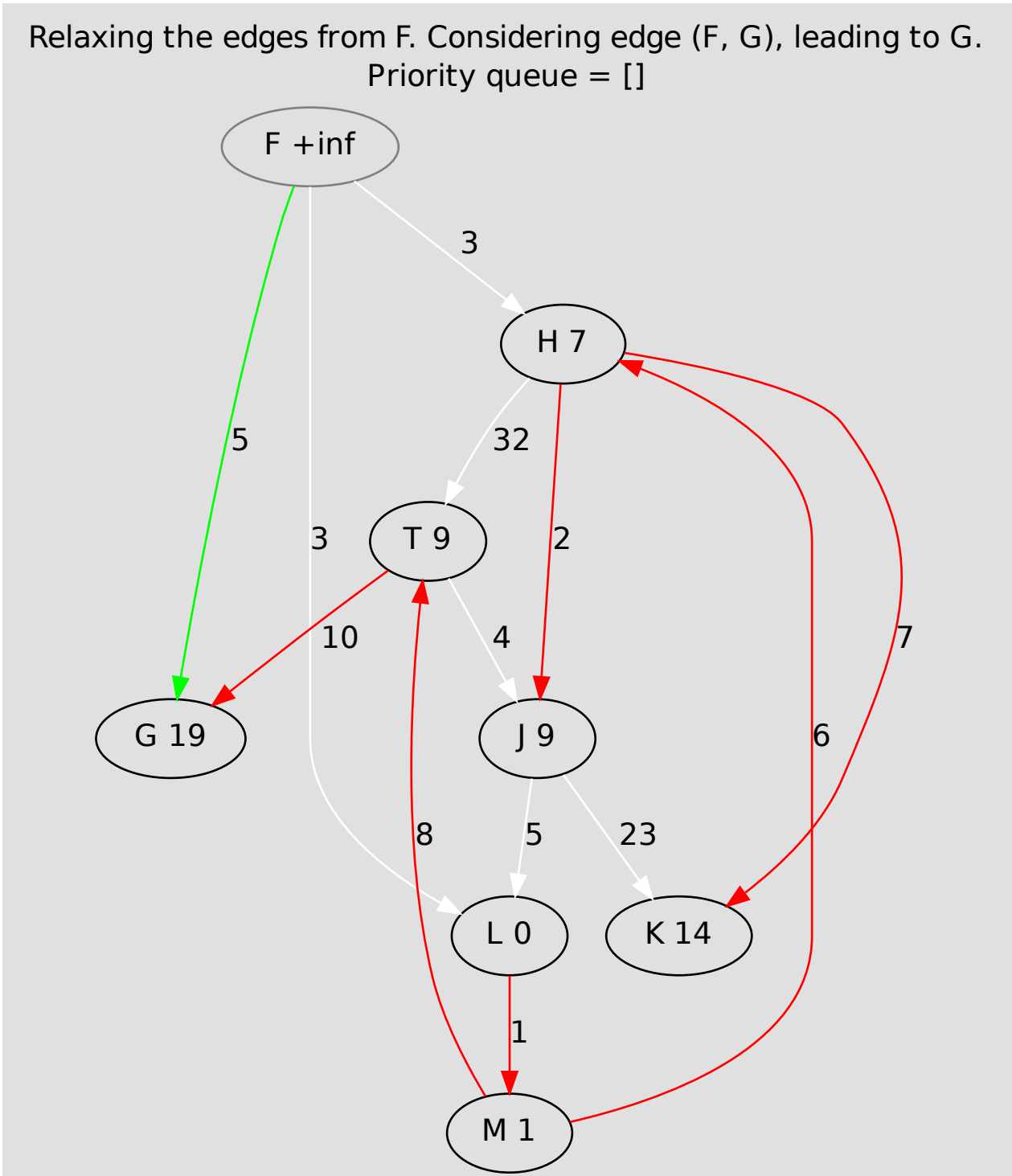

extractMin() -> F.
Priority queue = []
F +inf
H 7
T 9
G 19
J 9
L 0
K 14
M 1
3
5
32
3
2
10
4
7
6
8
5
23
1

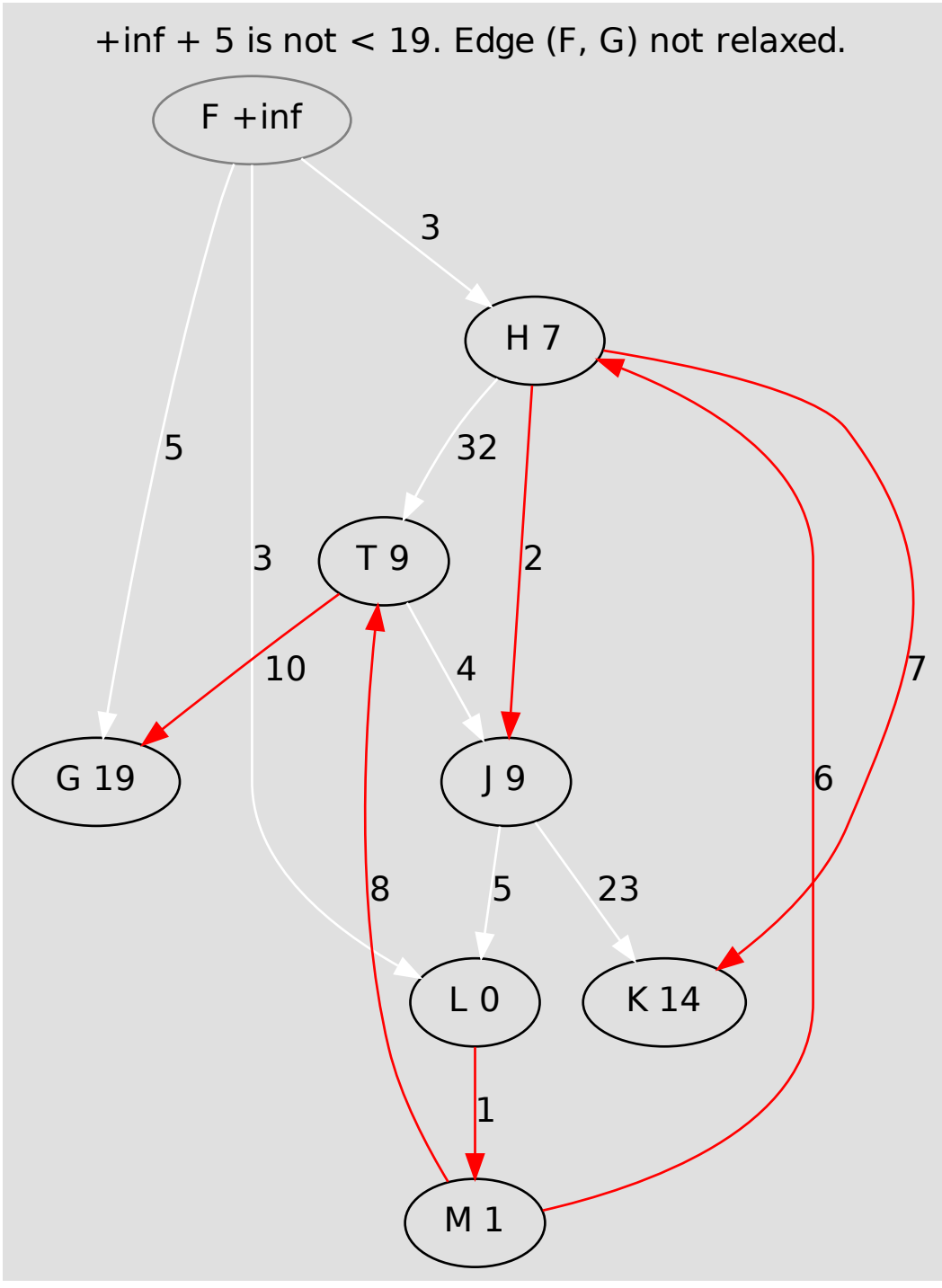

Relaxing the edges from F. Considering edge (F, H), leading to H.
Priority queue = []
F +inf
H 7
T 9
G 19
J 9
L 0
K 14
M 1
3
5
32
3
2
10
4
7
6
8
5
23
1

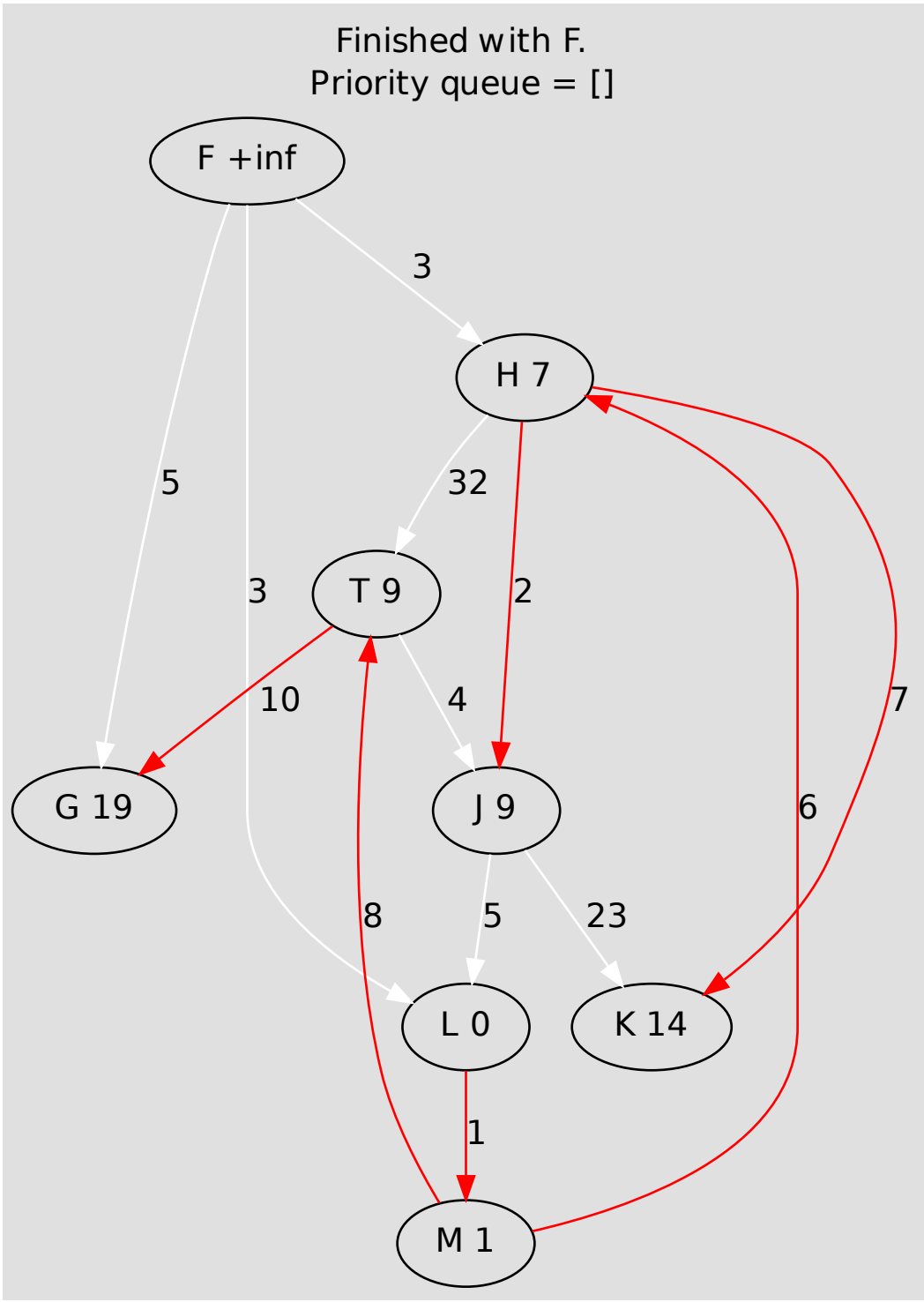

+inf + 3 is not < 7. Edge (F, H) not relaxed.
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1


Relaxing the edges from F. Considering edge (F, L), leading to L.
Priority queue = []
F +inf
H 7
T 9
G 19
J 9
L 0
K 14
M 1
3
5
32
3
2
10
4
7
6
8
5
23
1


+inf + 3 is not < 0. Edge (F, L) not relaxed.
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1


Relaxing the edges from F. Considering edge (F, G), leading to G.
Priority queue = []
F +inf
H 7
T 9
G 19
J 9
L 0
K 14
M 1
3
5
32
3
2
10
4
7
6
8
5
23
1


+inf + 5 is not < 19. Edge (F, G) not relaxed.
F +inf
3
H 7
5
32
3
T 9
2
10
4
7
G 19
J 9
6
8
5
23
L 0
K 14
1
M 1


Finished with F.
Priority queue = []
F +inf
H 7
T 9
G 19
J 9
L 0
K 14
M 1
3
5
32
3
2
10
4
7
6
8
5
23
1

Dijkstra single-source shortest paths
Generated by $Id: dijkstra.py 99 2010-11-18 10:48:22Z fms27 $

University of Cambridge Computer Laboratory