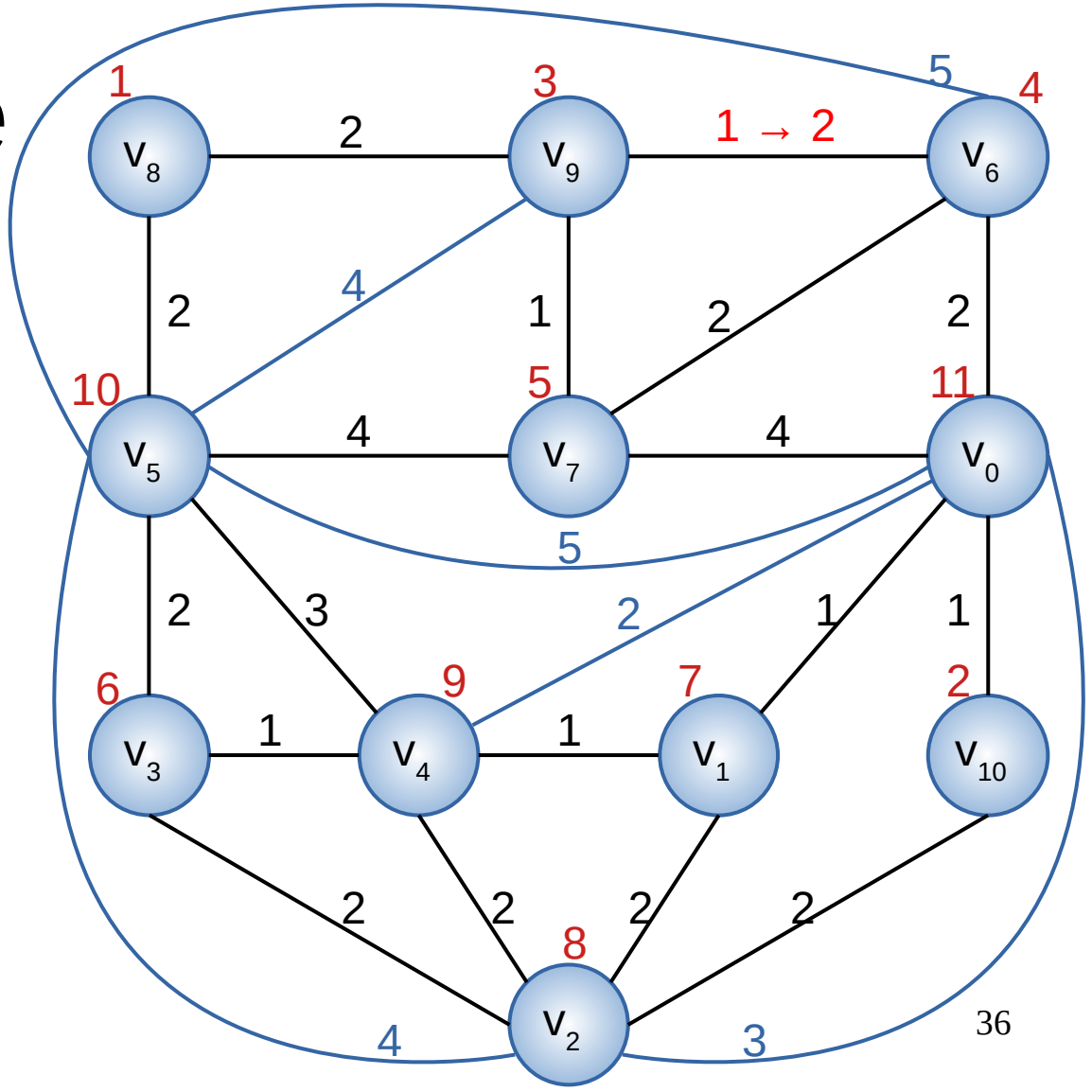
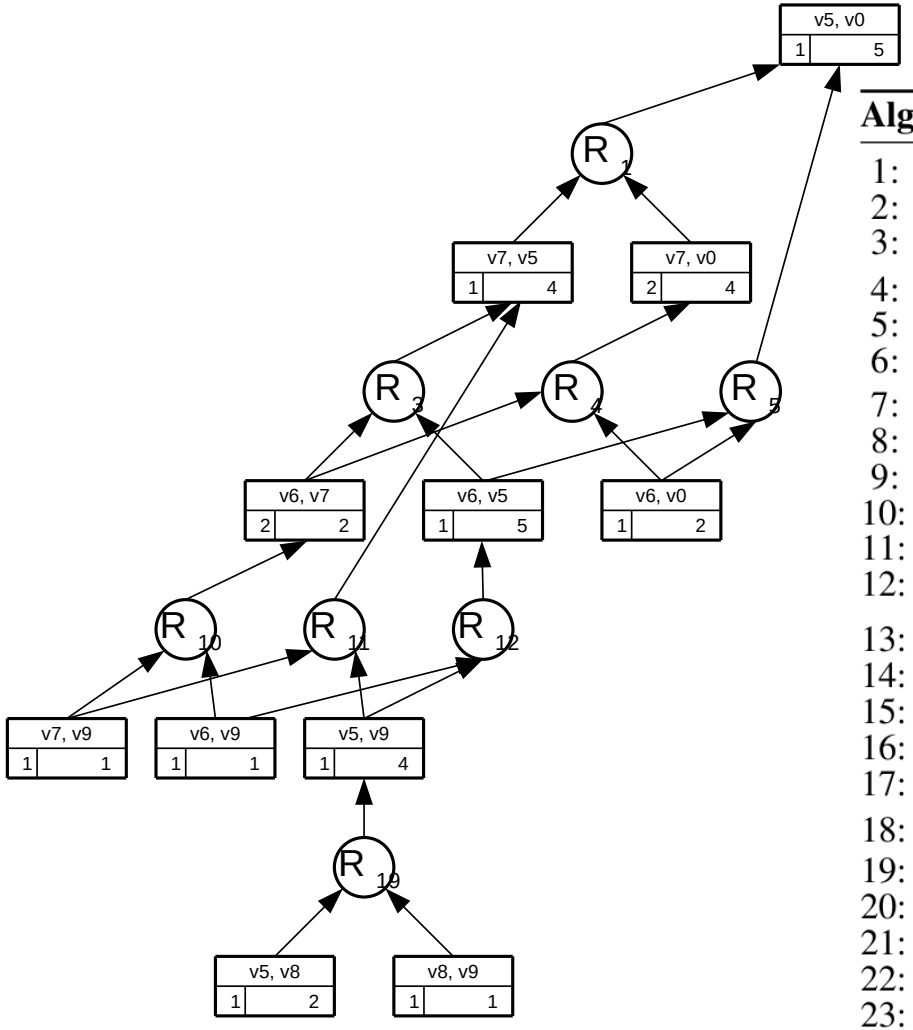
Weight Increase

- Weight decrease ($v_9 v_6$) = 2 → 1
- Implies further weight changes
- Let's use G* to find out which



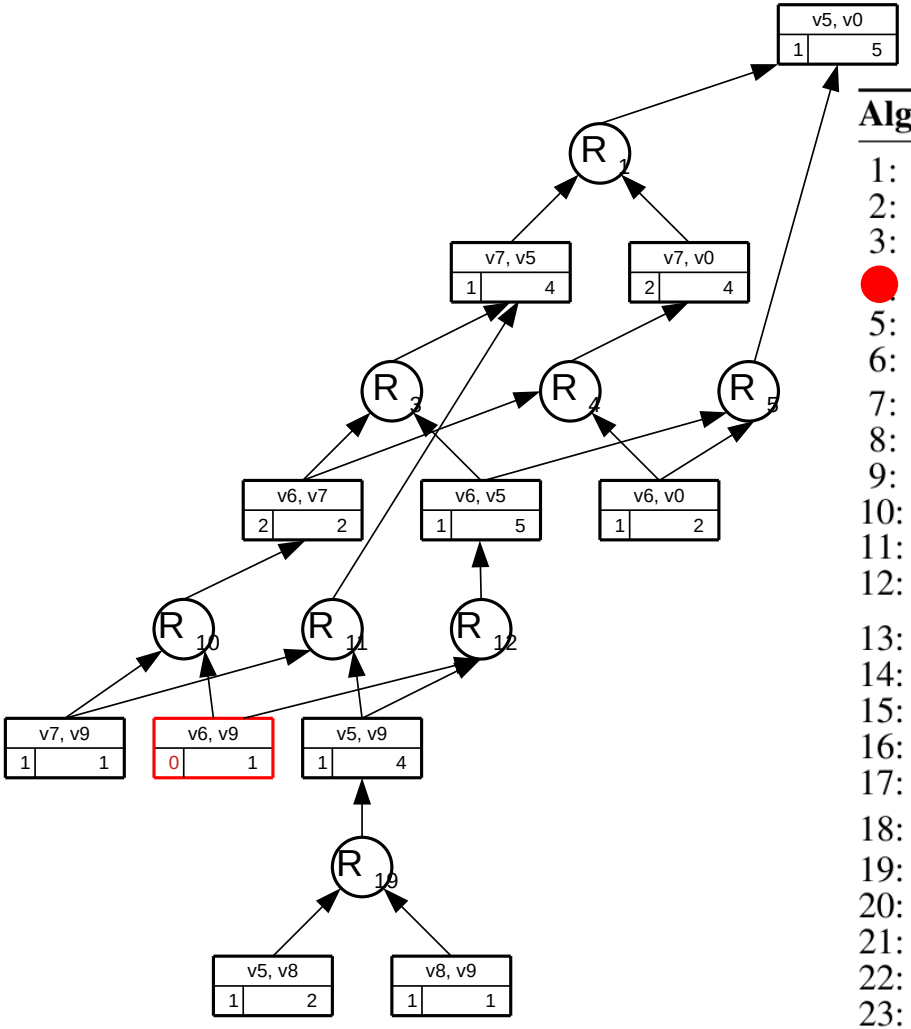


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

PriorityQueue $Q \leftarrow \emptyset$;
if $\phi(v_e, G^*) = \phi(e, G)$ **then**
  $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
$\phi(e, G) \leftarrow k$;
if $c_\phi(v_e) < 1$ **then**
  Q.push(v_e);
while $Q \neq \emptyset$ **do**
  $v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
  for each $v_r \in \mathsf{nbr}^+(v_s)$ **do**
    $v'_s \leftarrow \mathsf{nbr}^+(v_r)$;
    if $c_\phi(v'_s) < 1$ **then**
      if $v'_s \notin Q$ **then** Q.push(v'_s);

procedure updateWeight(G^*, v_s)
  for each $v_r \in \mathsf{nbr}^+(v_s)$ **do**
    $v_{s_1}, v_{s_2} \leftarrow \mathsf{nbr}^-(v_r)$; $v_{s_3} \leftarrow \mathsf{nbr}^+(v_r)$;
    if $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
      $c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
  $\phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
  for each $v_r \in \mathsf{nbr}^-(v_s)$ **do**
    $v_{s_1}, v_{s_2} \leftarrow \mathsf{nbr}^-(v_r)$;
    if $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
      $\phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
    else if $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
      $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
  $\phi(v_s, G^*) \leftarrow \phi$;

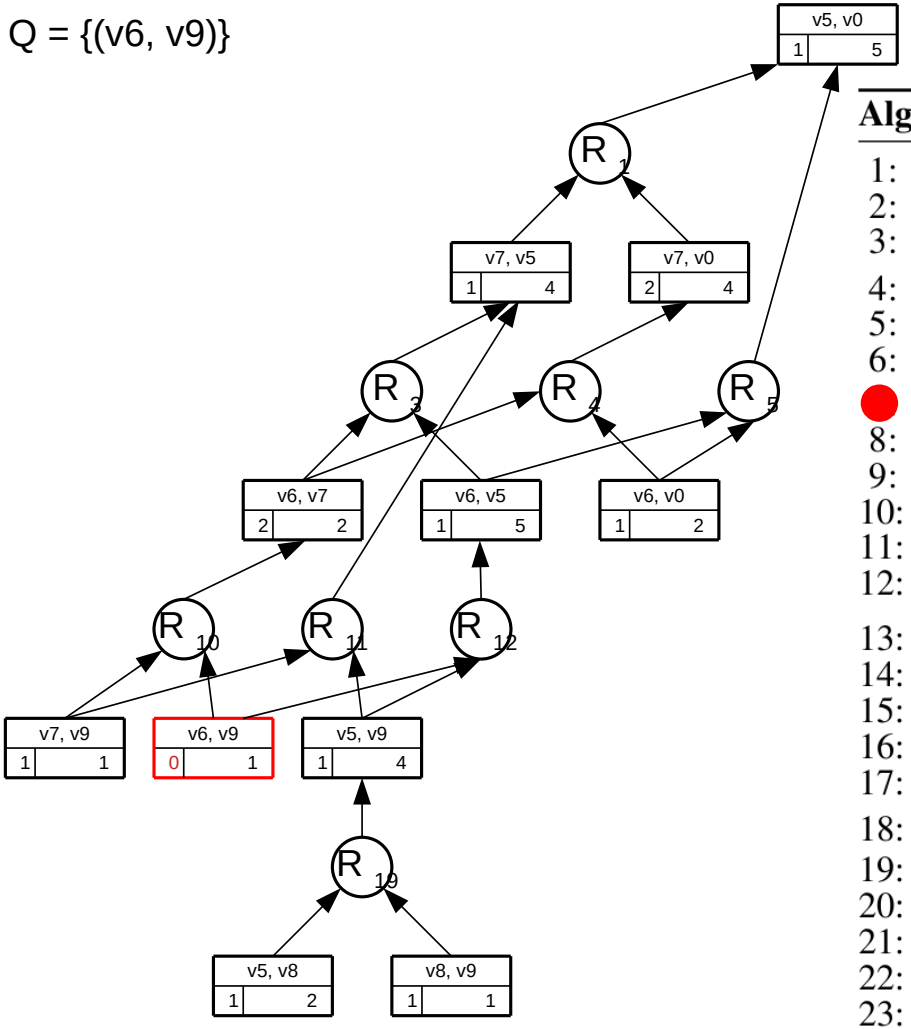


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: Q.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $v'_s \leftarrow \text{nbr}^+(v_r)$;
11: **if** $c_\phi(v'_s) < 1$ **then**
12: **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\phi(v_s, G^*) \leftarrow \phi$;

Q = {(v6, v9)}

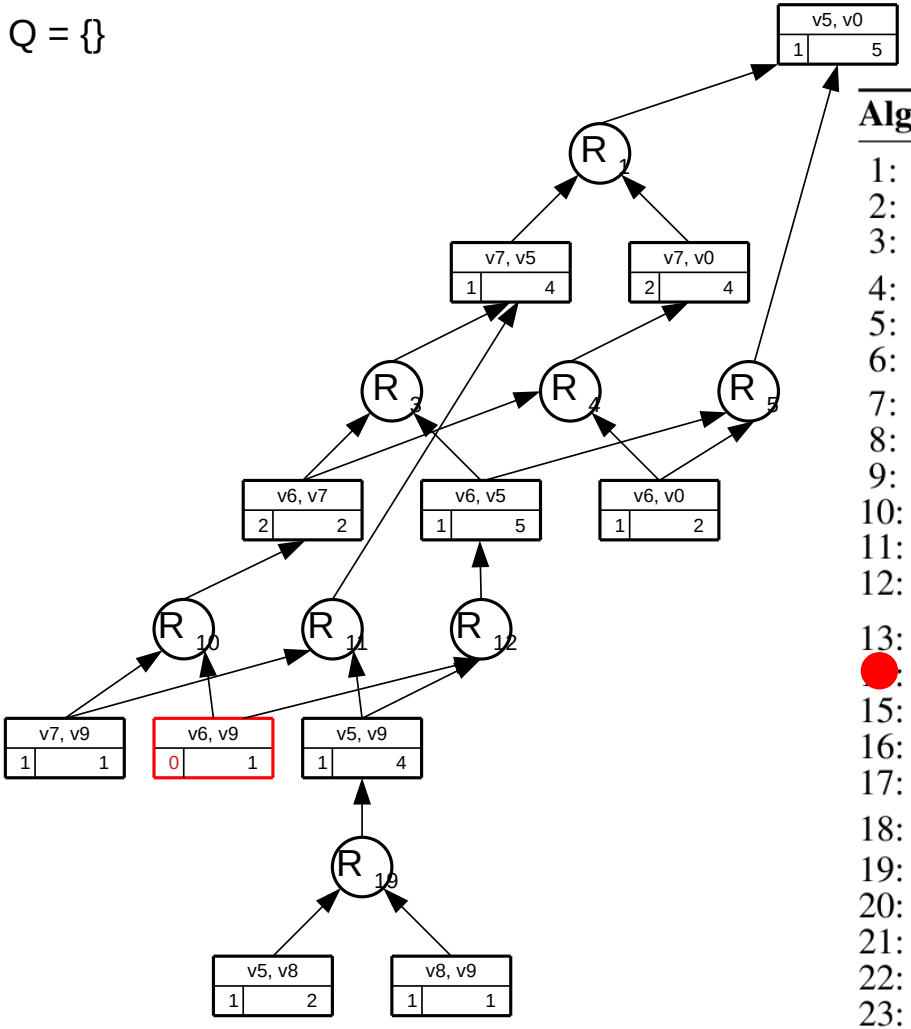


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

PriorityQueue $Q \leftarrow \emptyset$;
if $\phi(v_e, G^*) = \phi(e, G)$ **then**
  $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
$\phi(e, G) \leftarrow k$;
if $c_\phi(v_e) < 1$ **then**
  Q.push(v_e);
while $Q \neq \emptyset$ **do**
  $v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
  for each $v_r \in \text{nbr}^+(v_s)$ **do**
    $v'_s \leftarrow \text{nbr}^+(v_r)$;
    if $c_\phi(v'_s) < 1$ **then**
      if $v'_s \notin Q$ **then** Q.push(v'_s);
procedure updateWeight(G^*, v_s)
  for each $v_r \in \text{nbr}^+(v_s)$ **do**
    $v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
    if $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
      $c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
  $\phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
  for each $v_r \in \text{nbr}^-(v_s)$ **do**
    $v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
    if $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
      $\phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
    else if $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
      $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
  $\phi(v_s, G^*) \leftarrow \phi$;

Q = {}



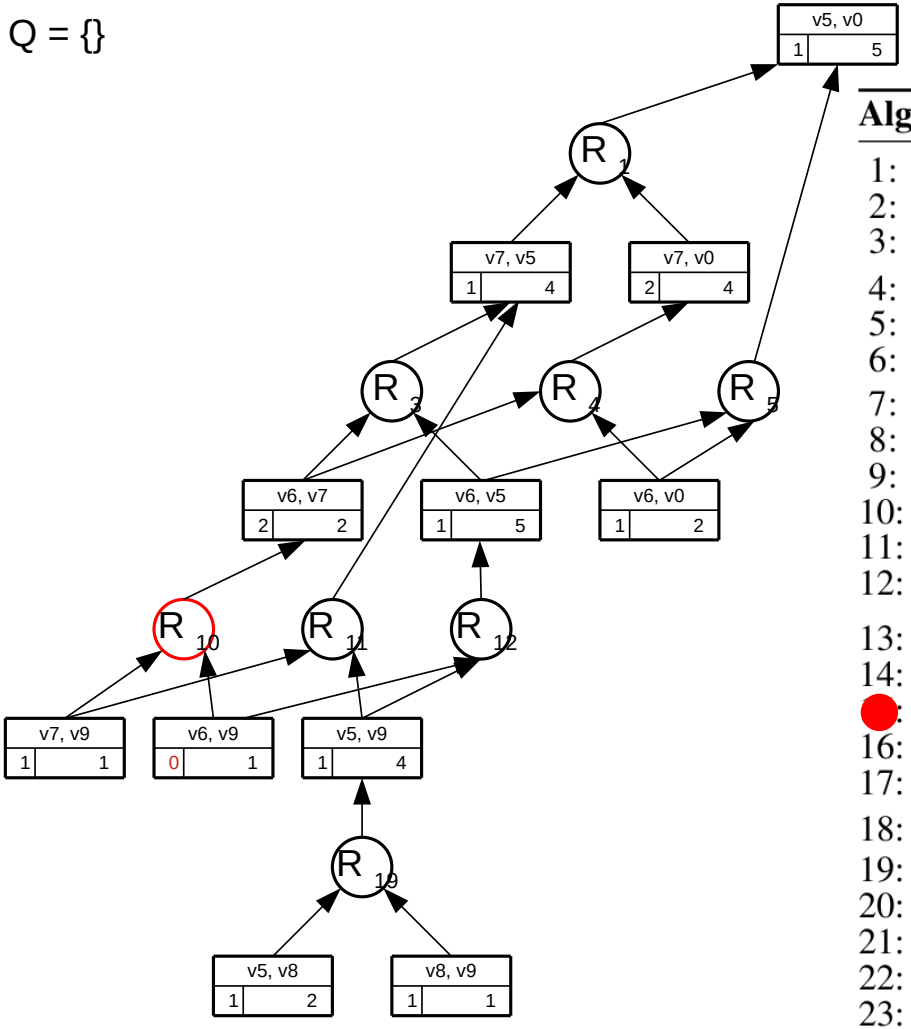
e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);

13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_s) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}

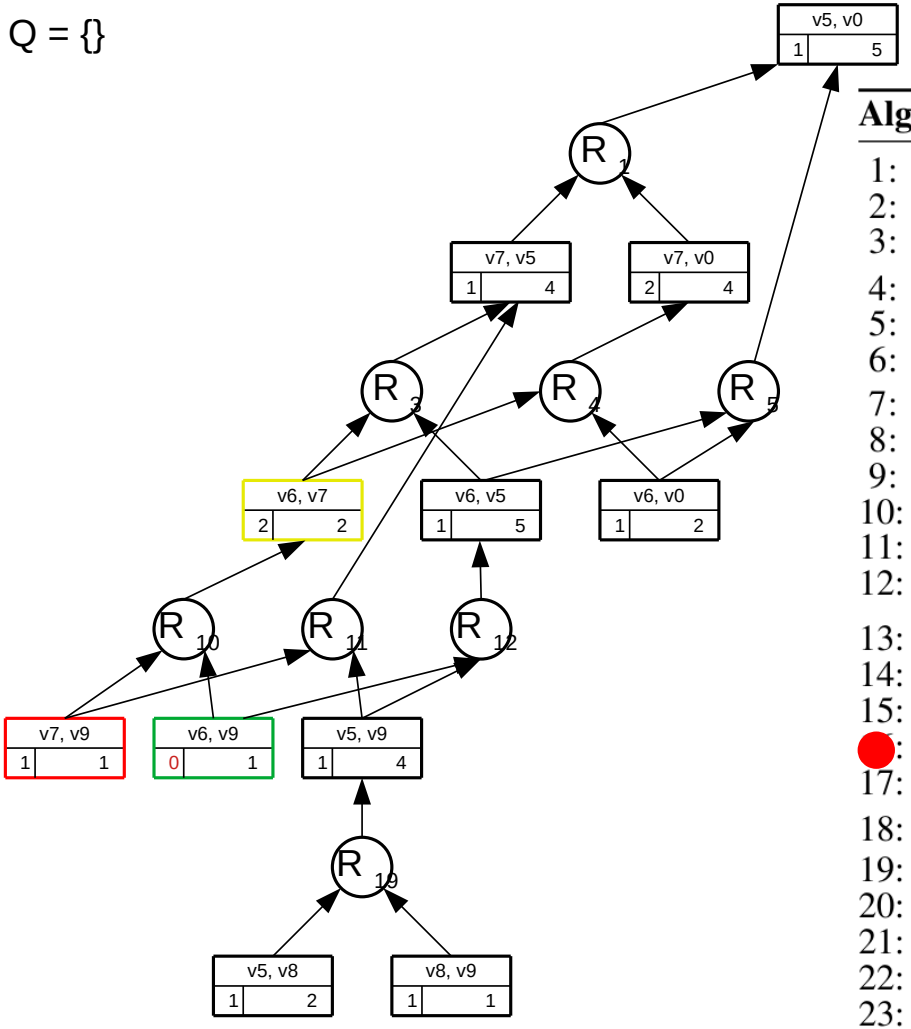


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

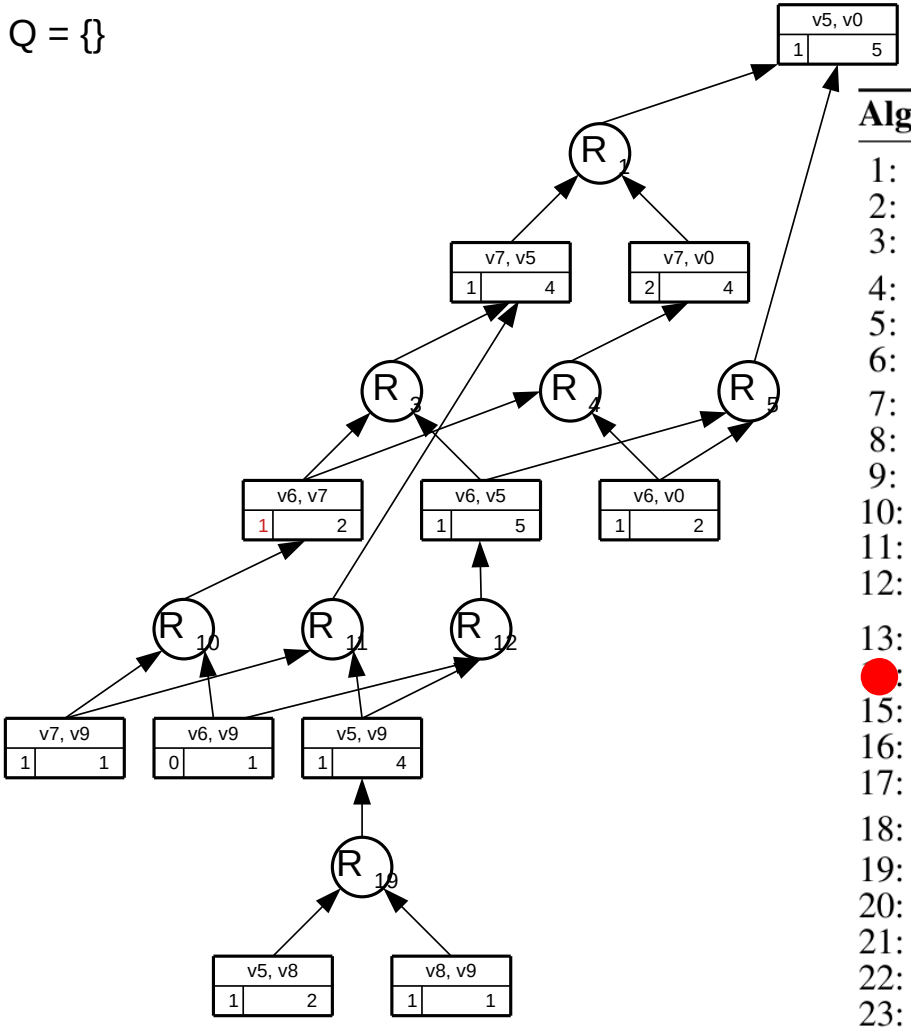


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

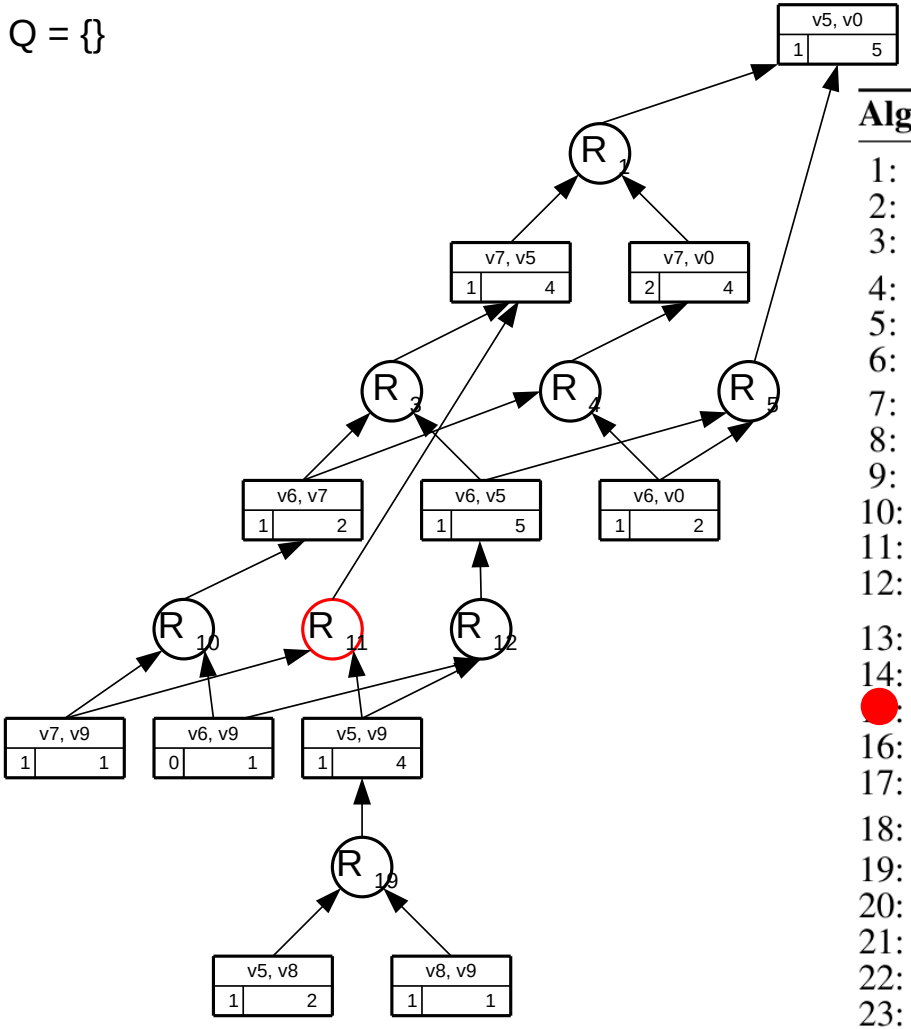


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

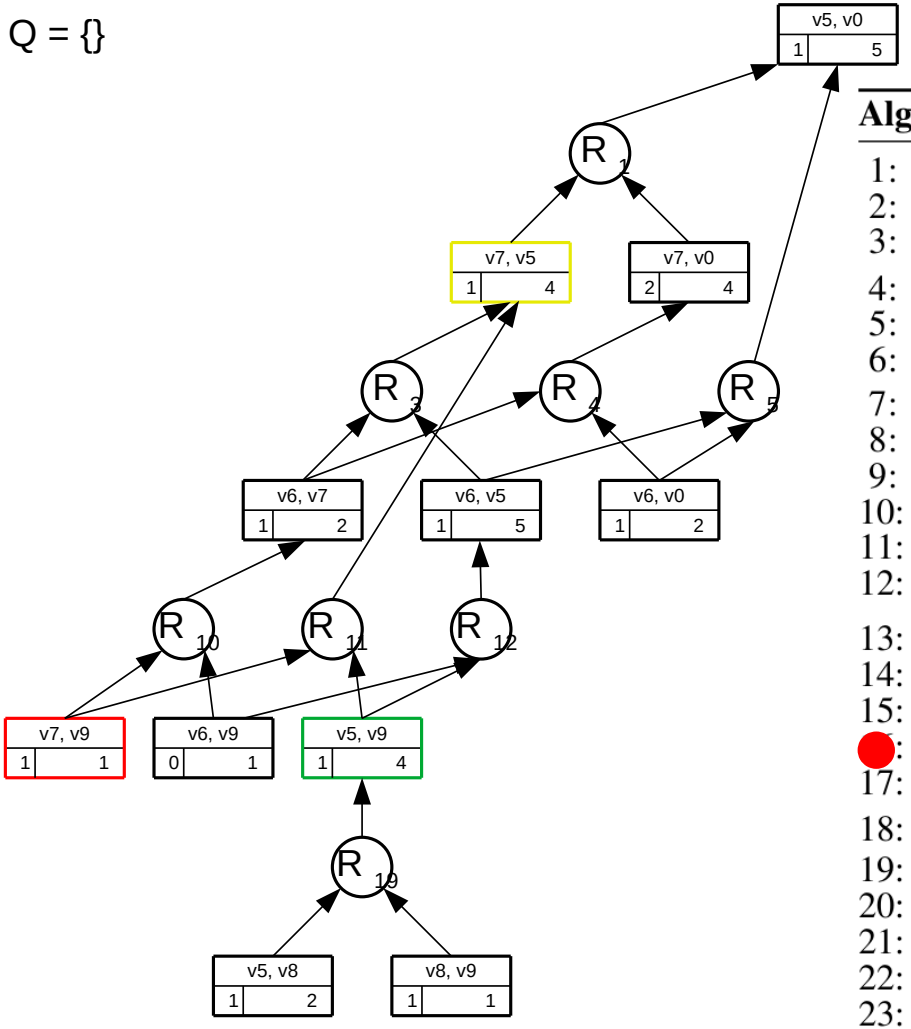


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

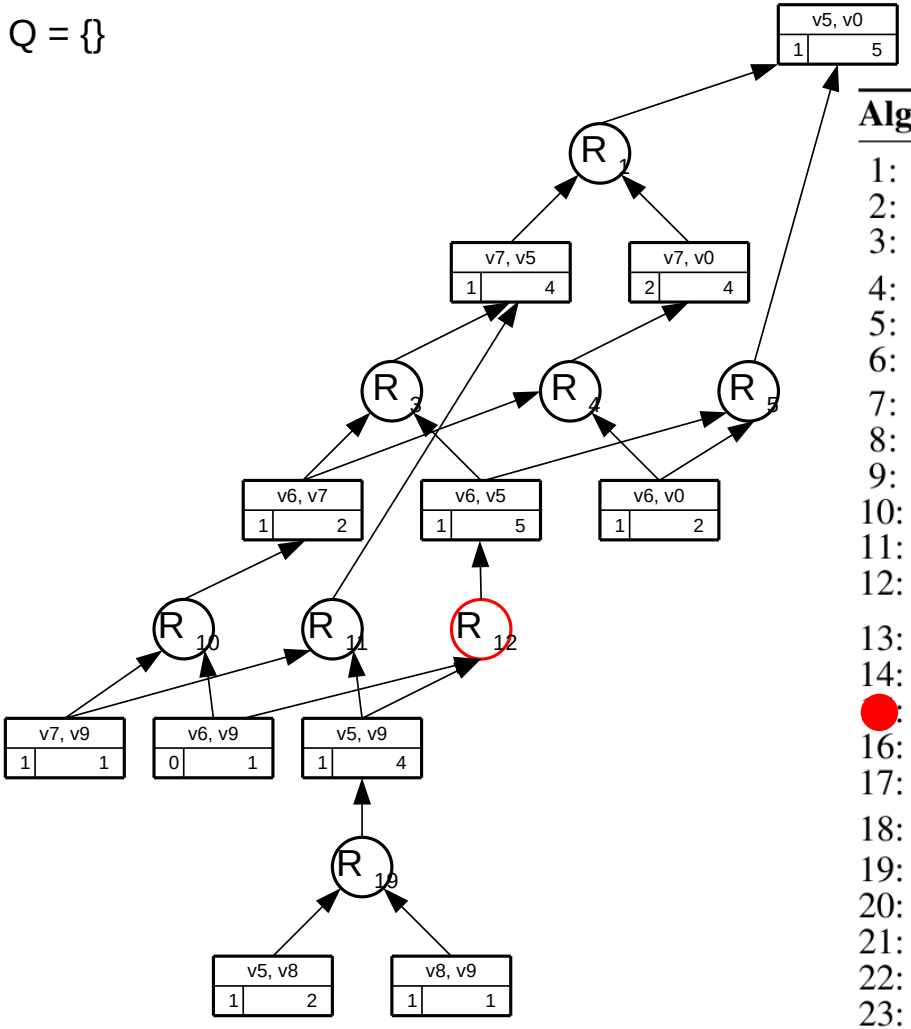


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: Q.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $v'_s \leftarrow \text{nbr}^+(v_r)$;
11: **if** $c_\phi(v'_s) < 1$ **then**
12: **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\phi(v_s, G^*) \leftarrow \phi$;

Q = {}

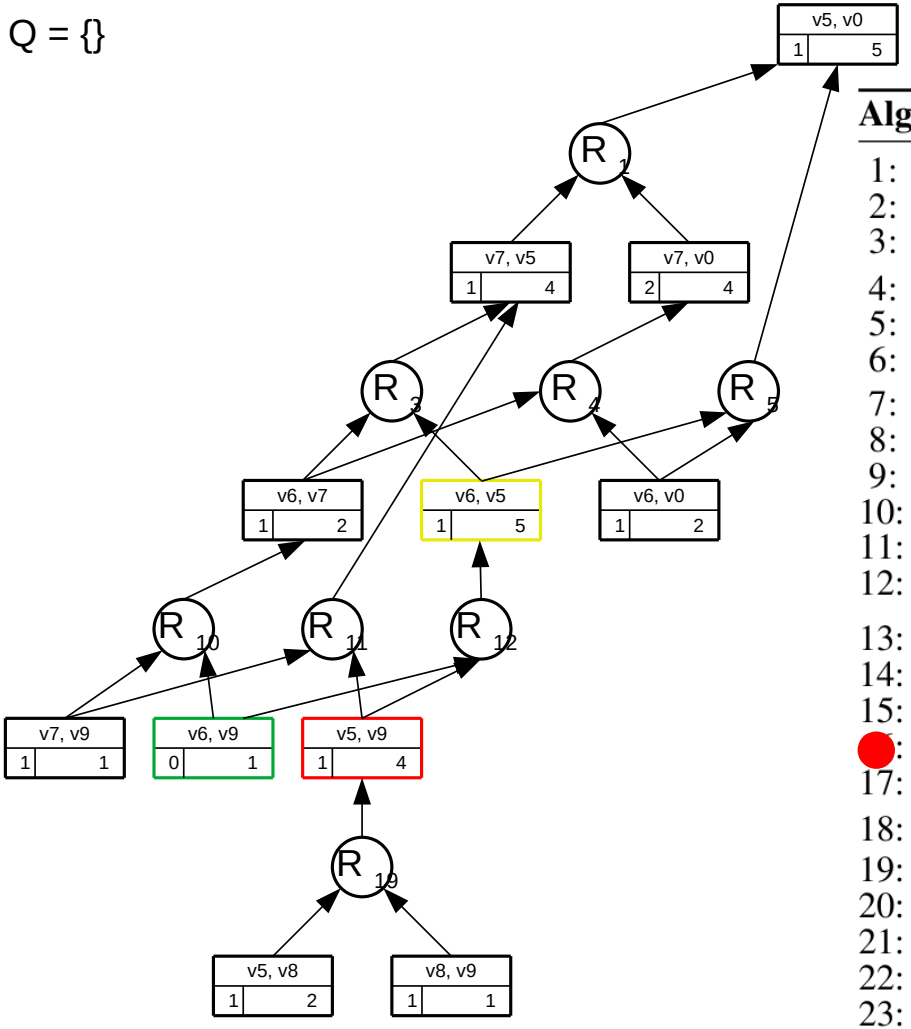


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

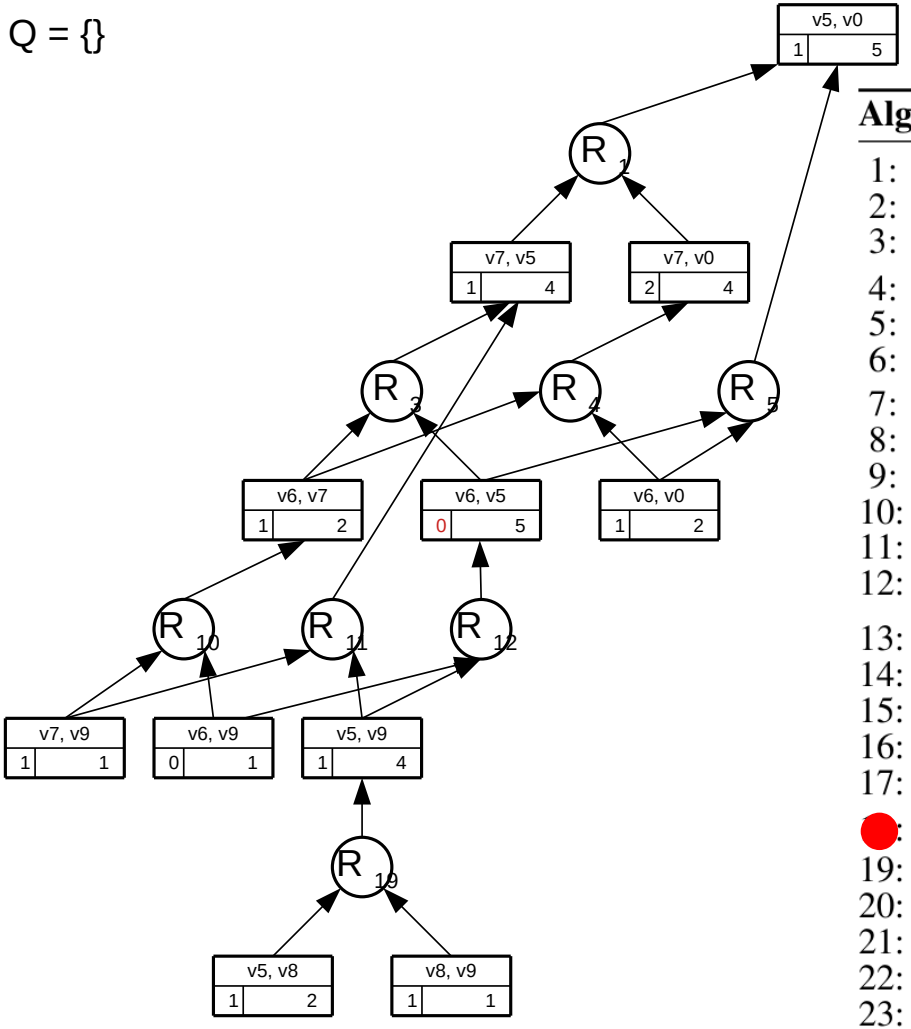


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

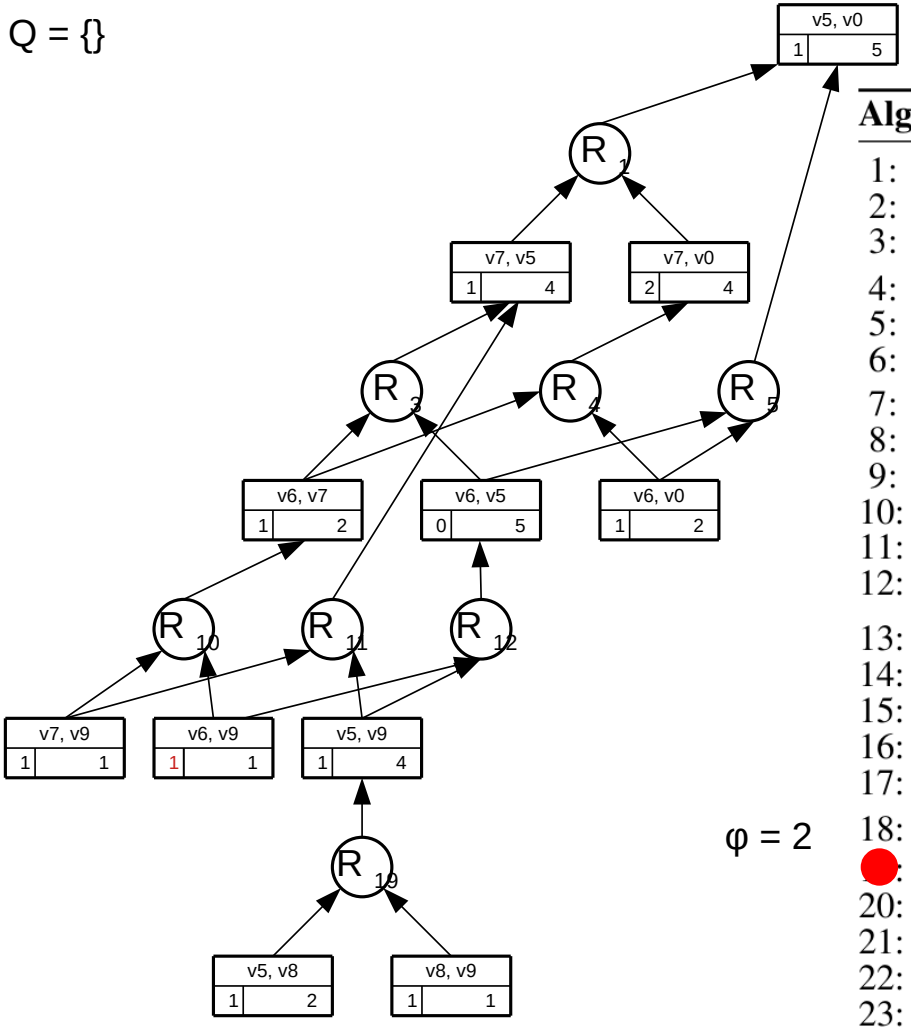


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}



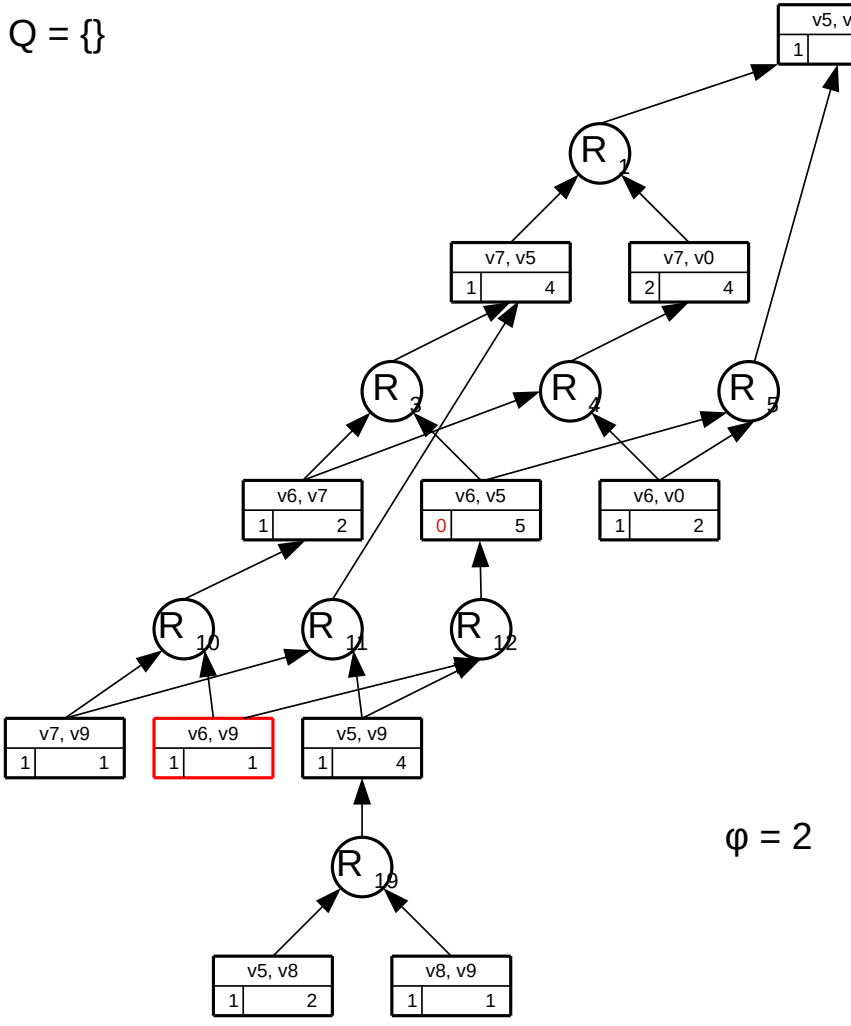
φ = 2

e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_s) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}



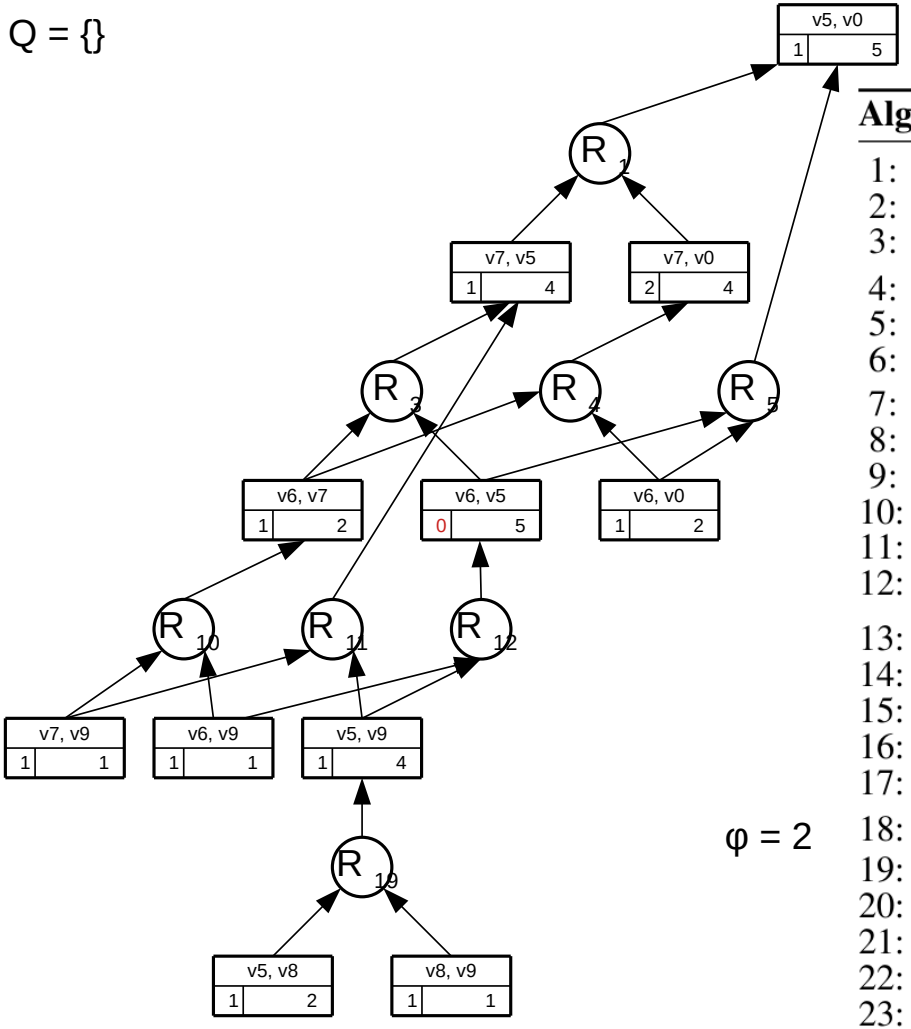
φ = 2

e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_s) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}



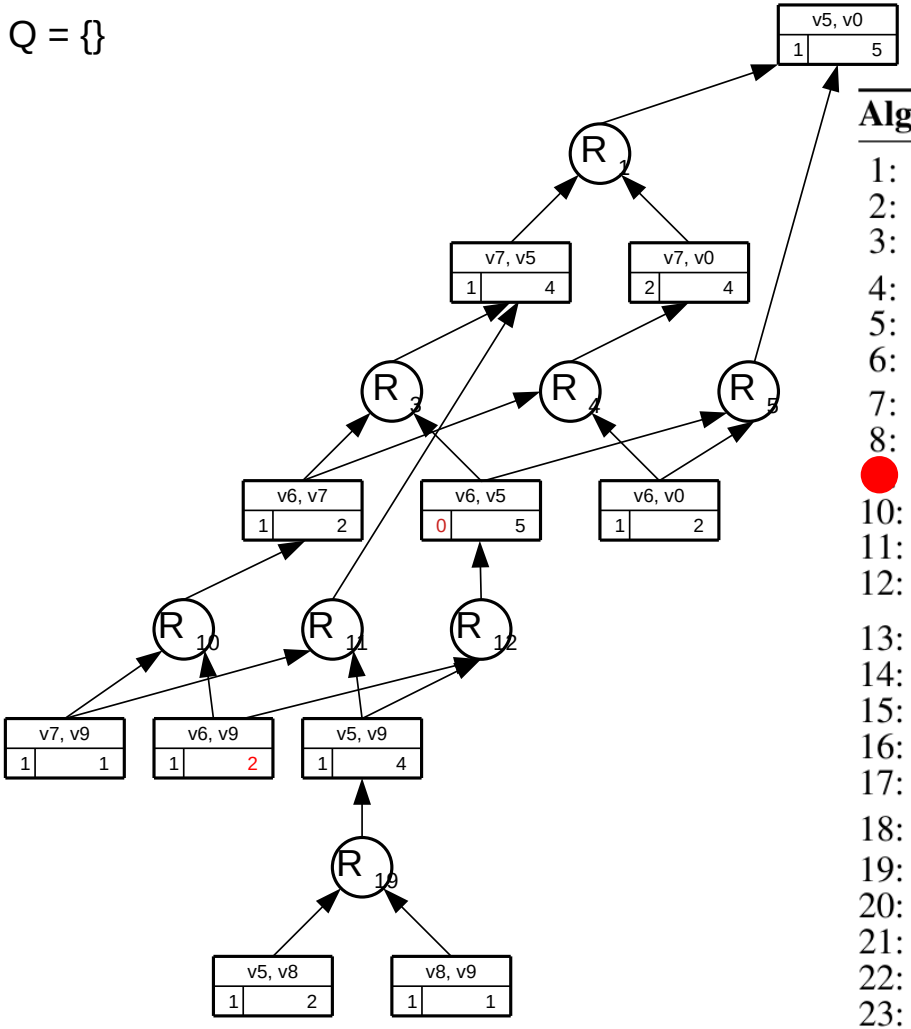
φ = 2

e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v_s');
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v_s' \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v_s') < 1$ **then**
12: $\quad\quad\quad$ **if** $v_s' \notin Q$ **then** Q.push(v_s');
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do** nbr$^-$(v$_s$) = {}
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
$\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

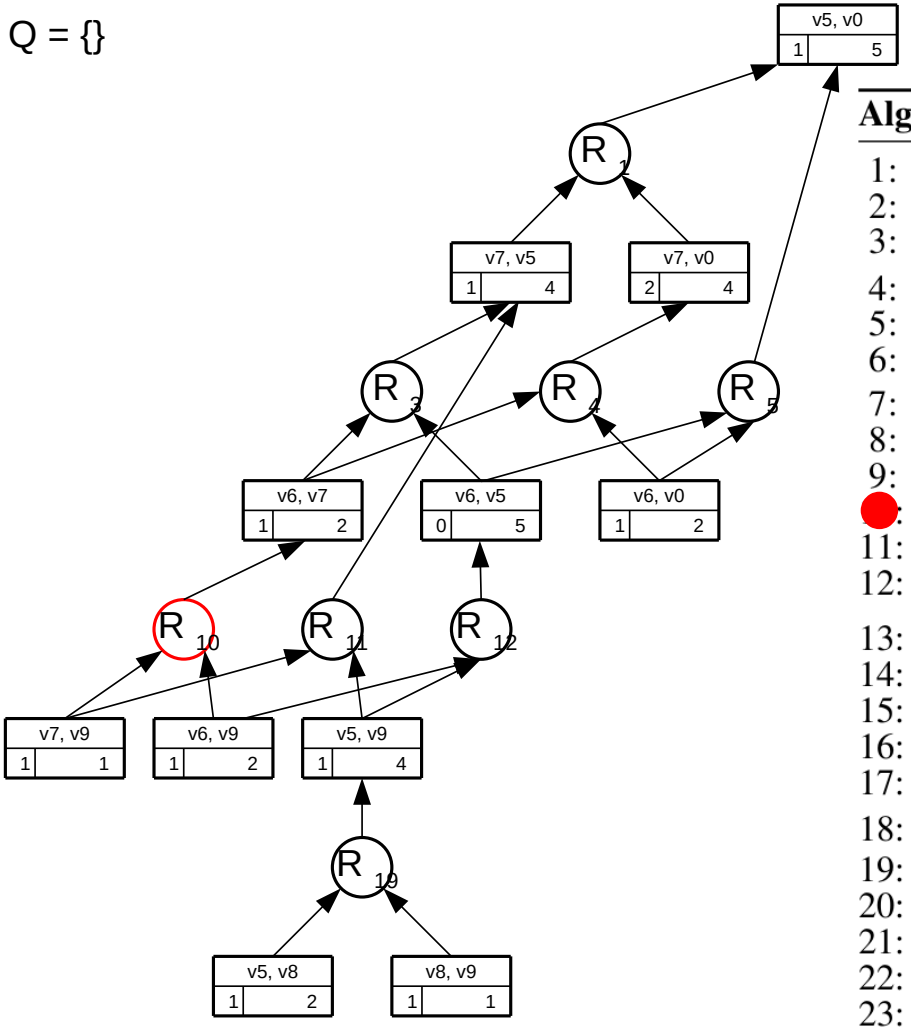


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v_s');
$\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v_s' \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v_s') < 1$ **then**
12: $\quad\quad\quad$ **if** $v_s' \notin Q$ **then** Q.push(v_s');
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

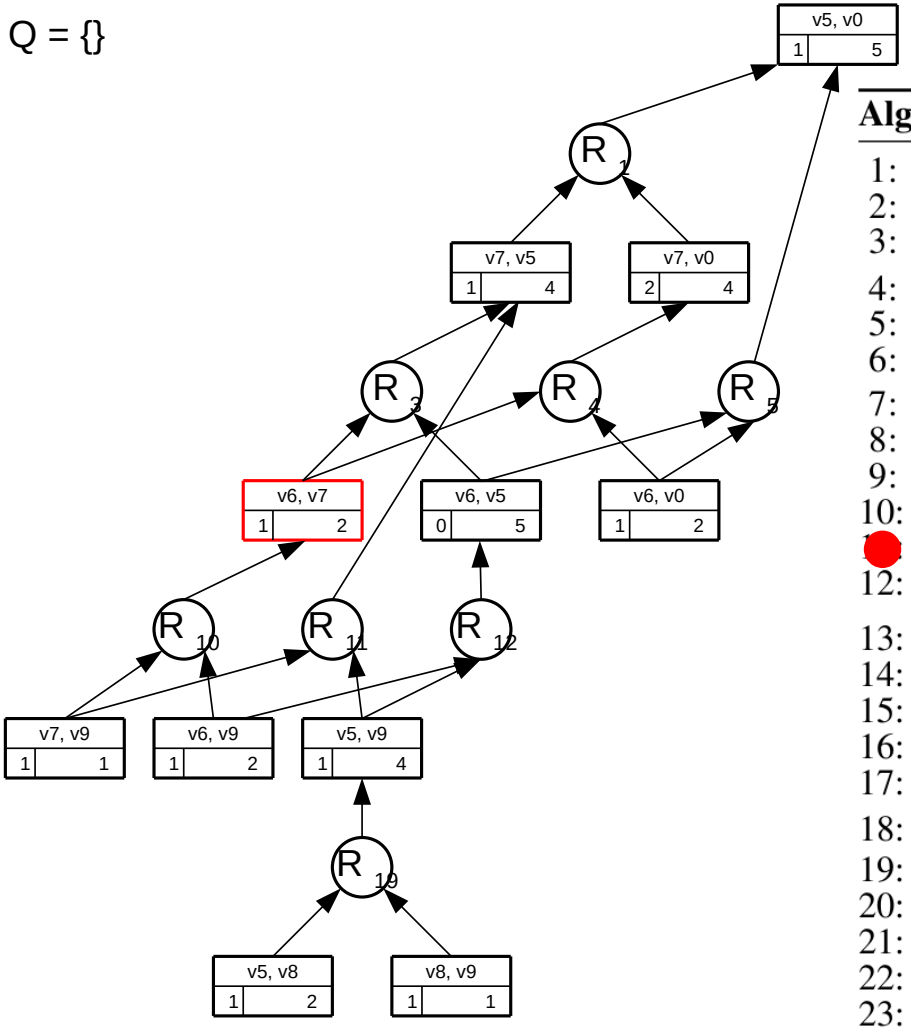


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_s) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}

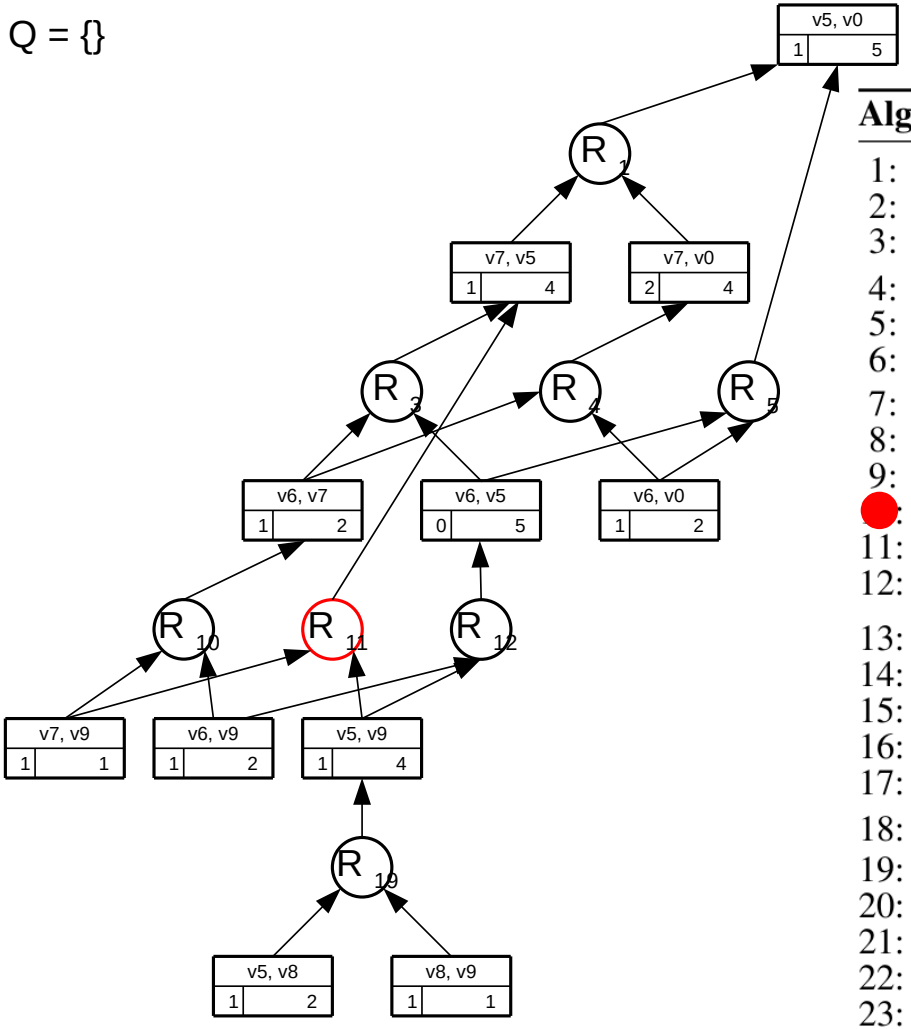


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_s) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}

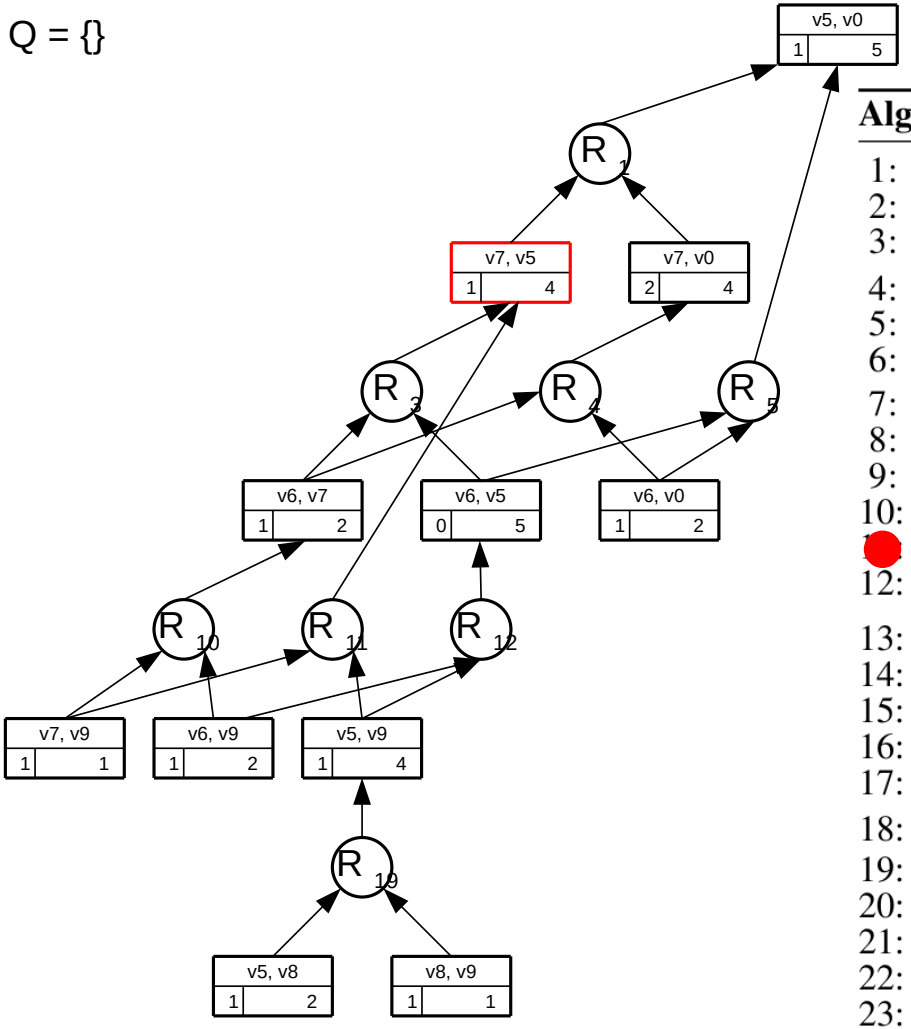


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \mathsf{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \mathsf{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \mathsf{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \mathsf{nbr}^-(v_r)$; $v_{s_3} \leftarrow \mathsf{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \mathsf{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \mathsf{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

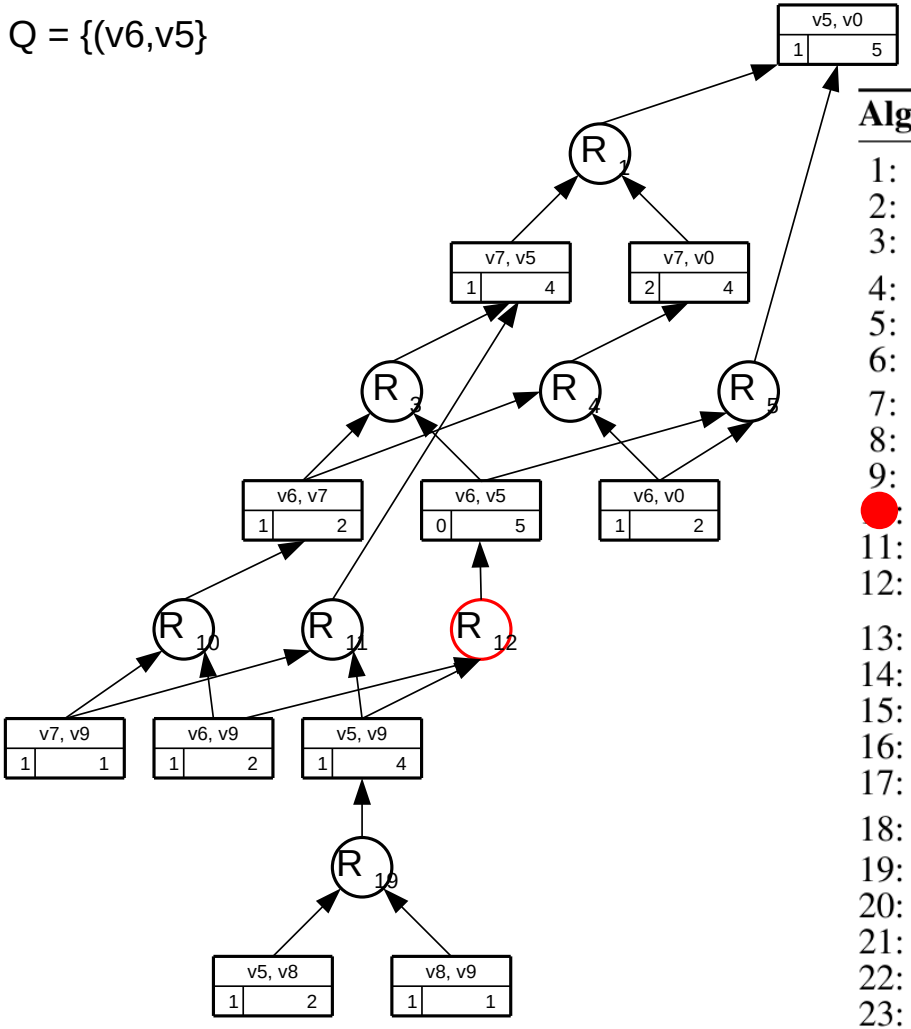


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {(v6,v5}

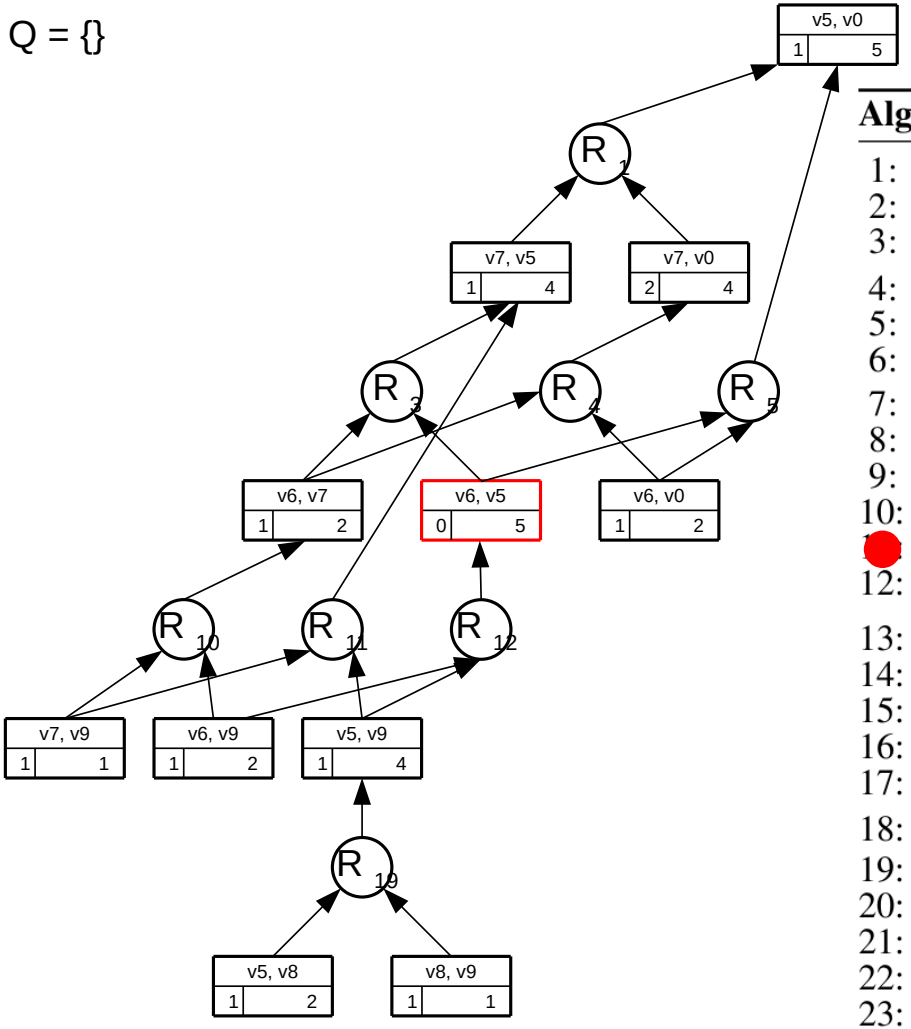


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \mathsf{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \mathsf{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \mathsf{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \mathsf{nbr}^-(v_r)$; $v_{s_3} \leftarrow \mathsf{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \mathsf{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \mathsf{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

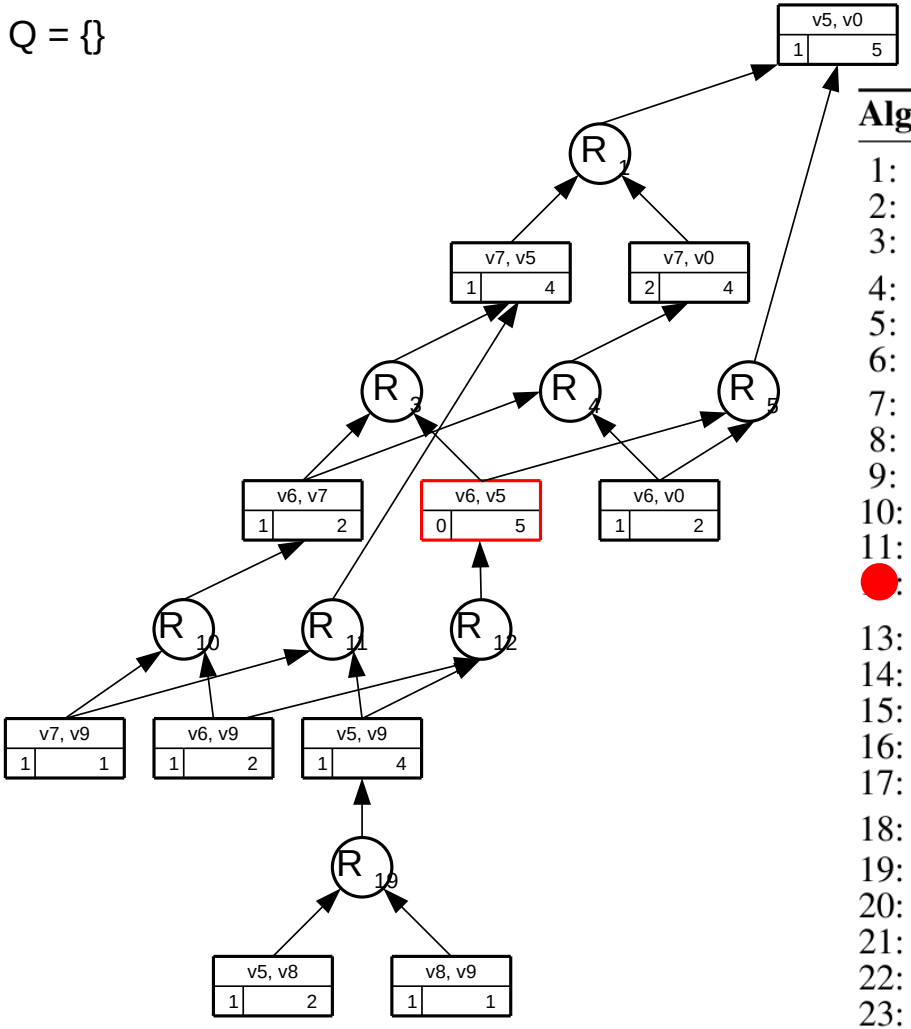


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

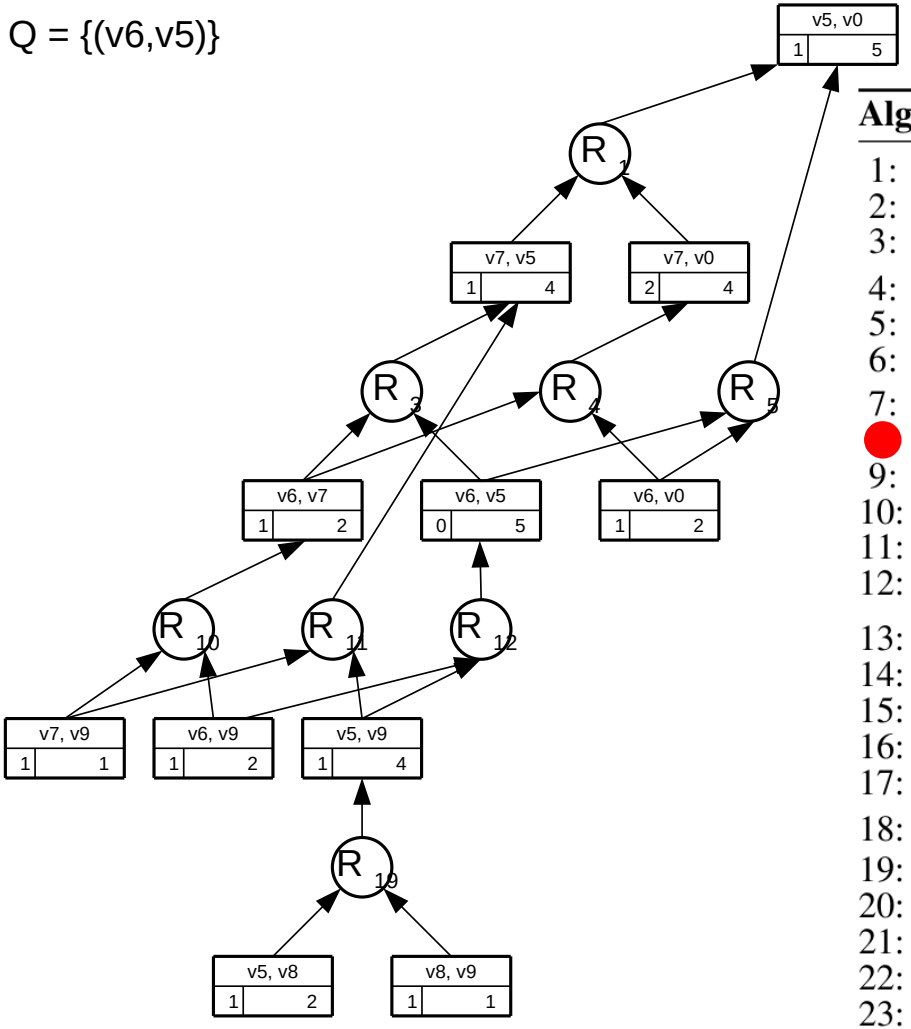


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {(v6,v5)}

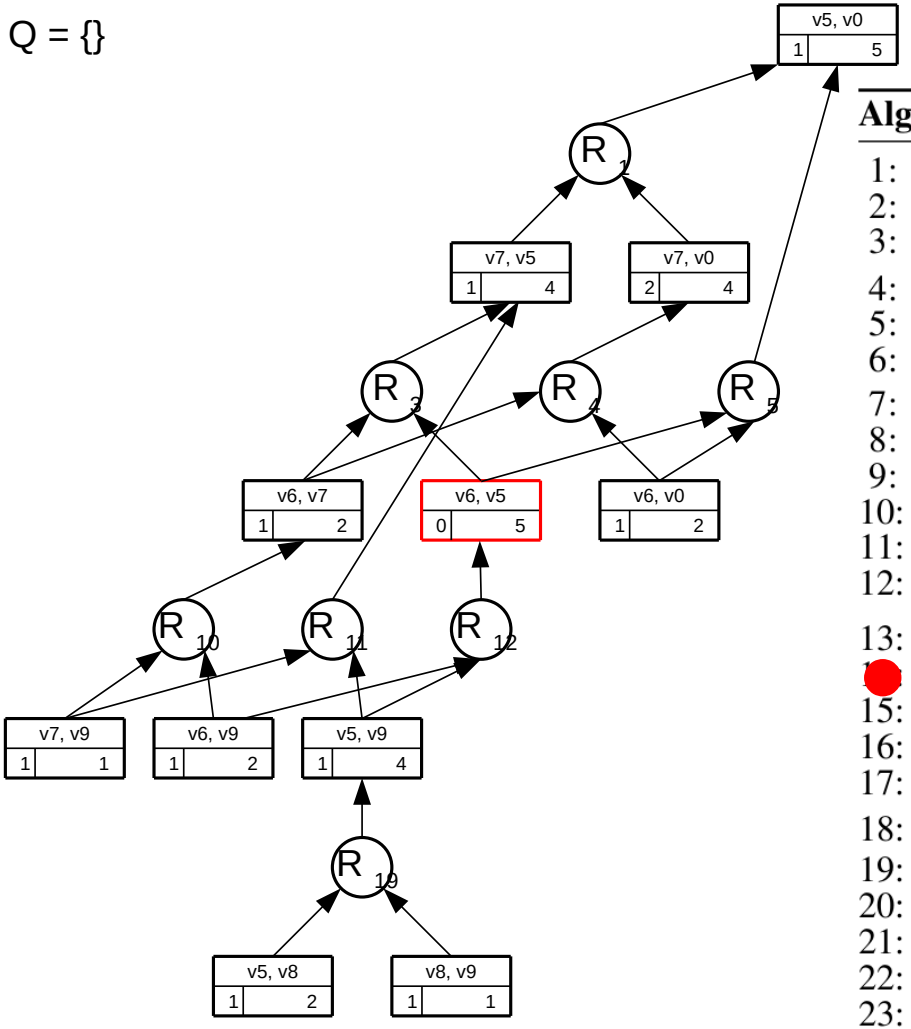


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_s) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}

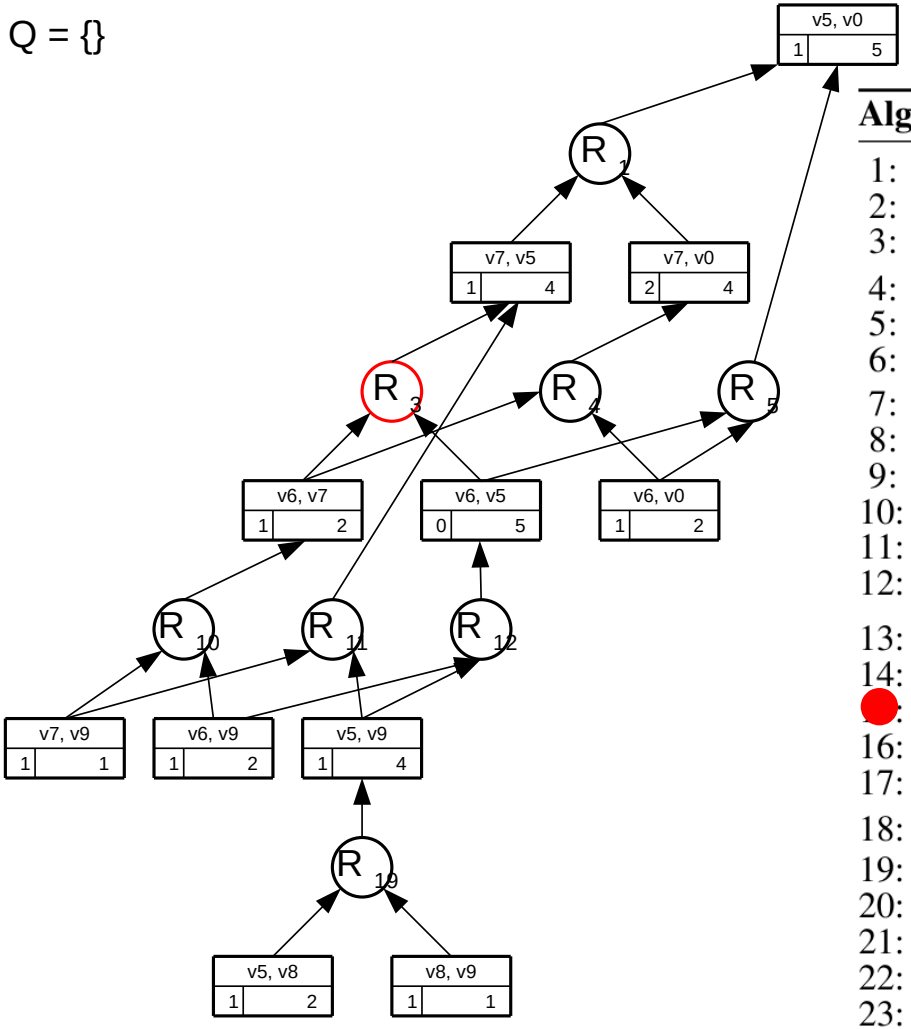


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

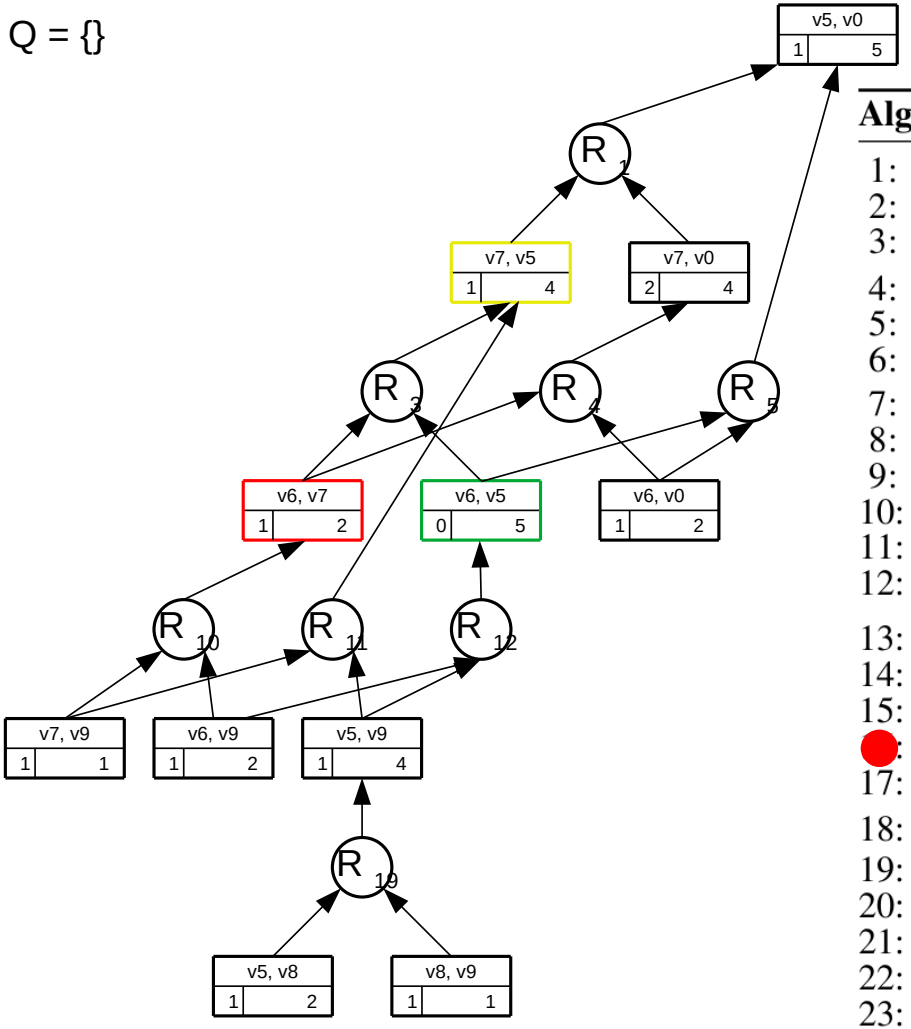


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}

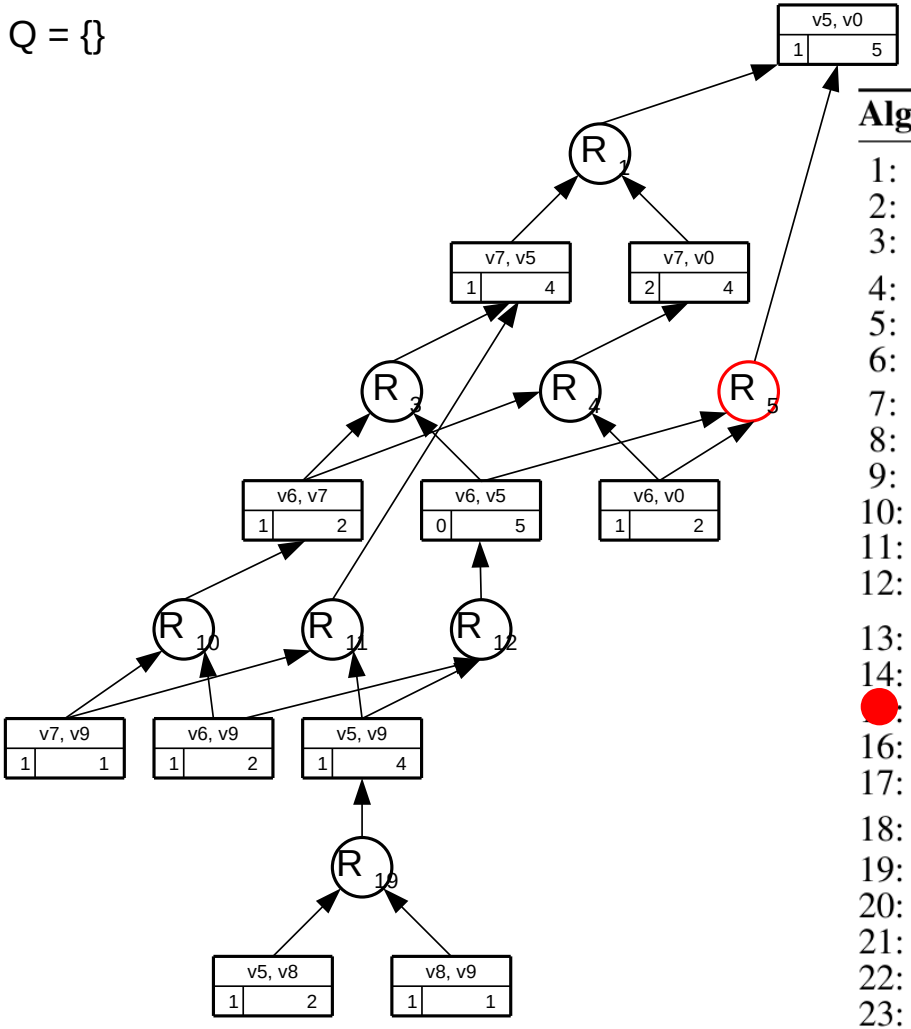


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_s) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}

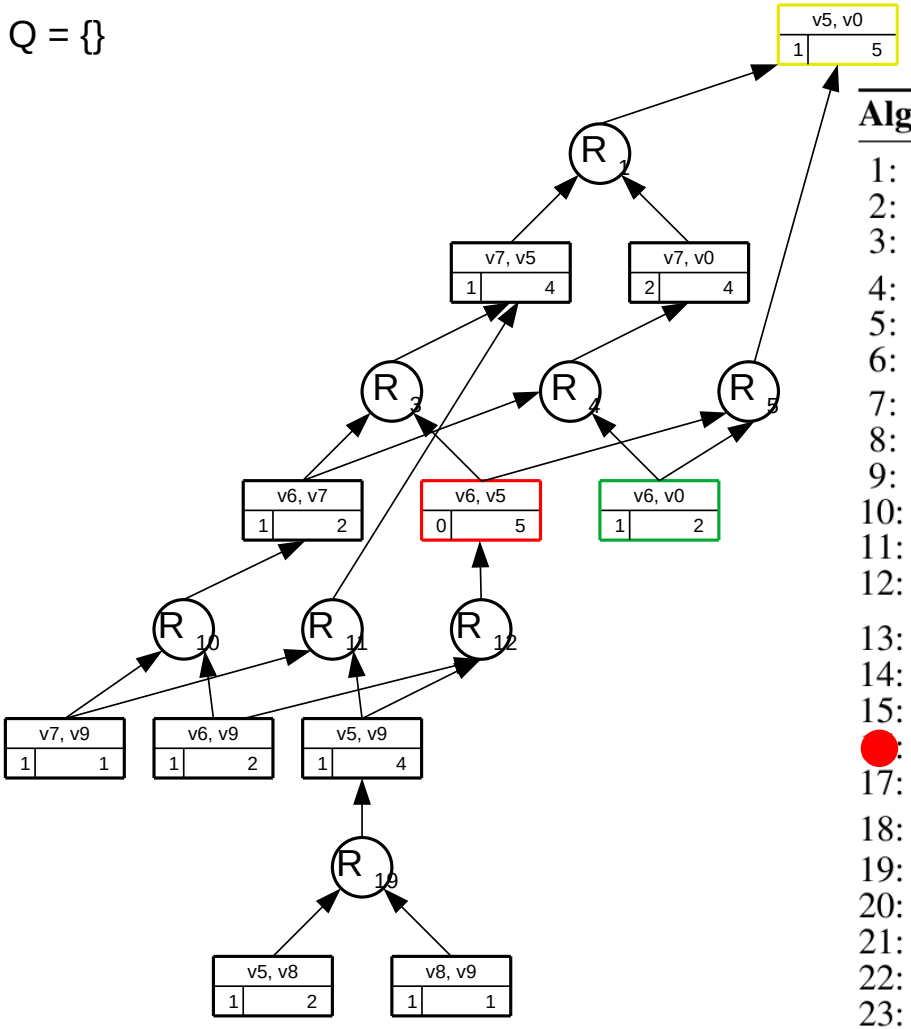


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_s) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}

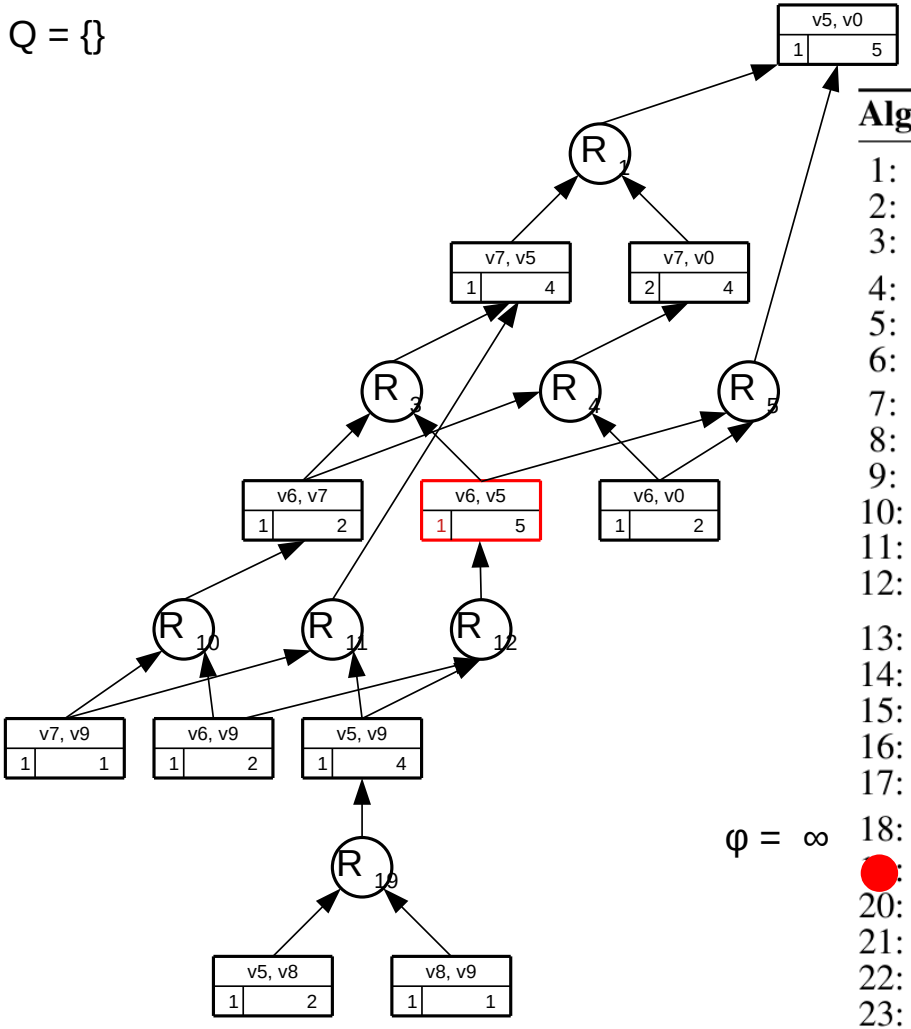


e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

1: PriorityQueue $Q \leftarrow \emptyset$;
2: **if** $\phi(v_e, G^*) = \phi(e, G)$ **then**
3: $\quad c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$;
4: $\phi(e, G) \leftarrow k$;
5: **if** $c_\phi(v_e) < 1$ **then**
6: $\quad Q$.push(v_e);
7: **while** $Q \neq \emptyset$ **do**
8: $\quad v_s \leftarrow Q$.pop(); updateWeight(G^*, v'_s);
9: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
10: $\quad\quad v'_s \leftarrow \text{nbr}^+(v_r)$;
11: $\quad\quad$ **if** $c_\phi(v'_s) < 1$ **then**
12: $\quad\quad\quad$ **if** $v'_s \notin Q$ **then** Q.push(v'_s);
13: **procedure** updateWeight(G^*, v_s)
14: $\quad$ **for each** $v_r \in \text{nbr}^+(v_s)$ **do**
15: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$; $v_{s_3} \leftarrow \text{nbr}^+(v_r)$;
16: $\quad\quad$ **if** $\phi(v_{s_3}, G^*) = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
17: $\quad\quad\quad c_\phi(v_{s_3}) \leftarrow c_\phi(v_{s_3}) - 1$;
18: $\quad \phi \leftarrow \phi(s, G)$; $c_\phi(v_s) \leftarrow 1$;
19: $\quad$ **for each** $v_r \in \text{nbr}^-(v_s)$ **do**
20: $\quad\quad v_{s_1}, v_{s_2} \leftarrow \text{nbr}^-(v_r)$;
21: $\quad\quad$ **if** $\phi > \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
22: $\quad\quad\quad \phi \leftarrow \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$; $c_\phi(v_s) \leftarrow 1$;
23: $\quad\quad$ **else if** $\phi = \phi(v_{s_1}, G^*) + \phi(v_{s_2}, G^*)$ **then**
24: $\quad\quad\quad c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$;
25: $\quad \phi(v_s, G^*) \leftarrow \phi$;

Q = {}



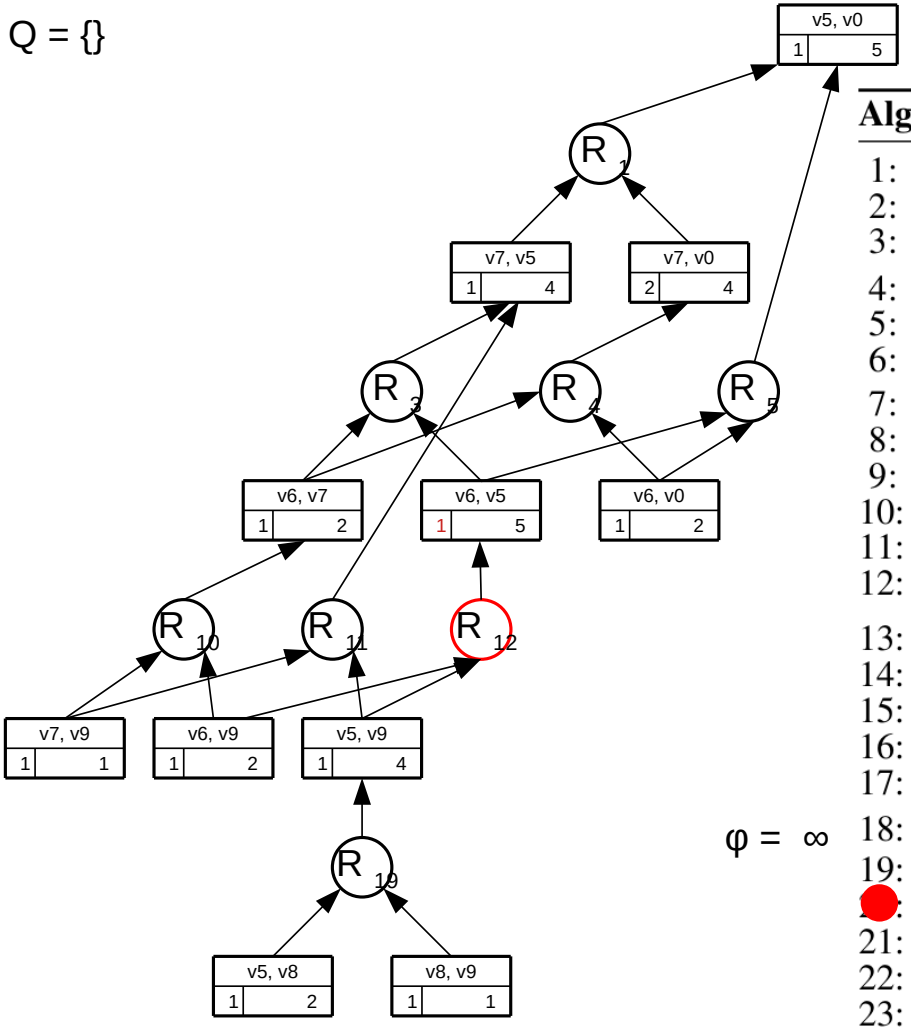
φ = ∞

e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_s) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}



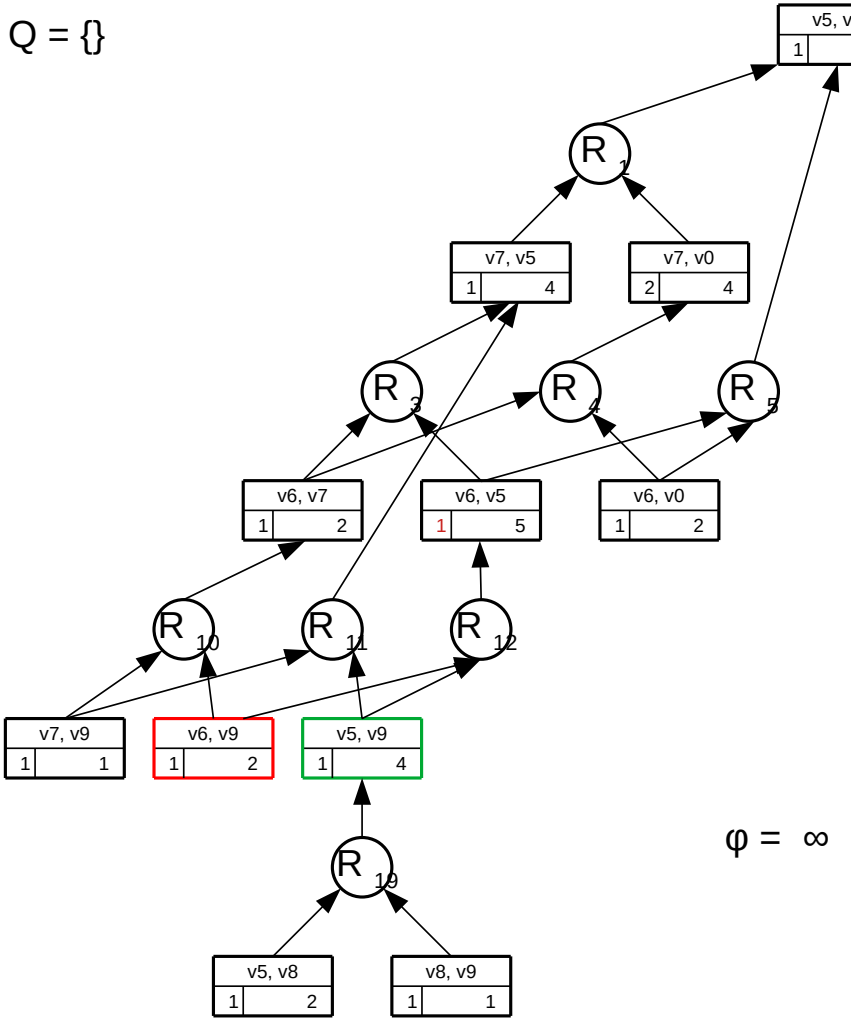
φ = ∞

e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_s) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}



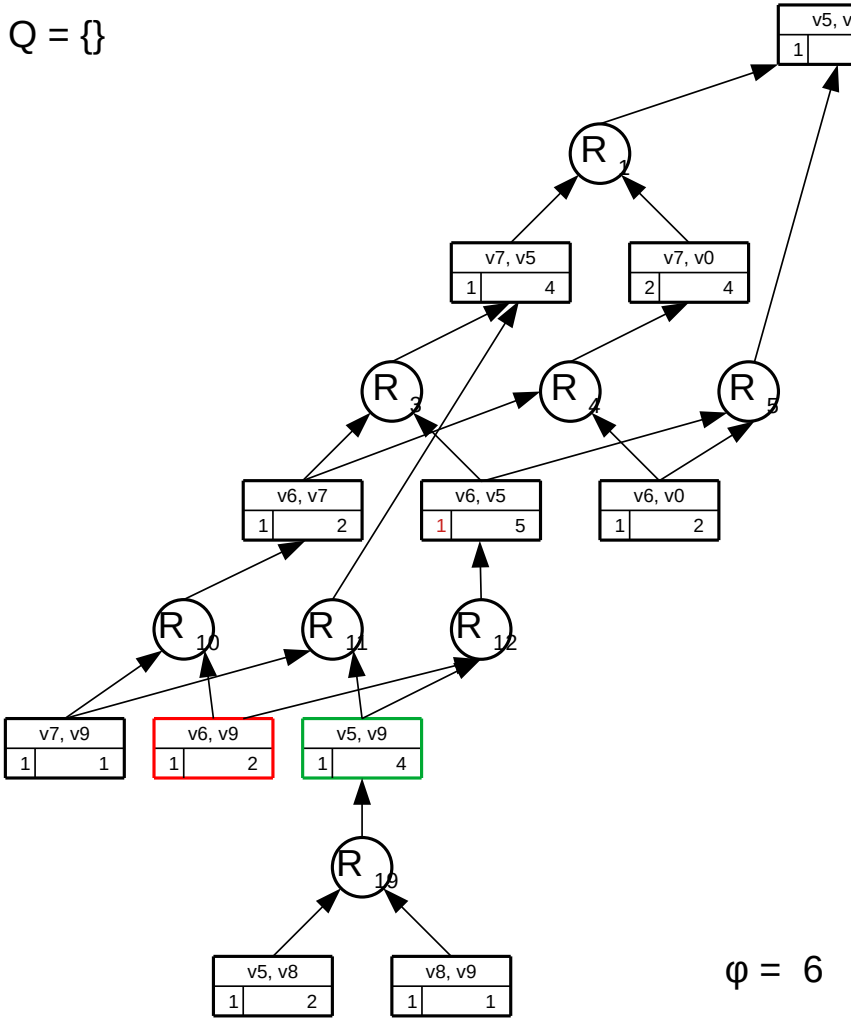
φ = ∞

e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_r) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}



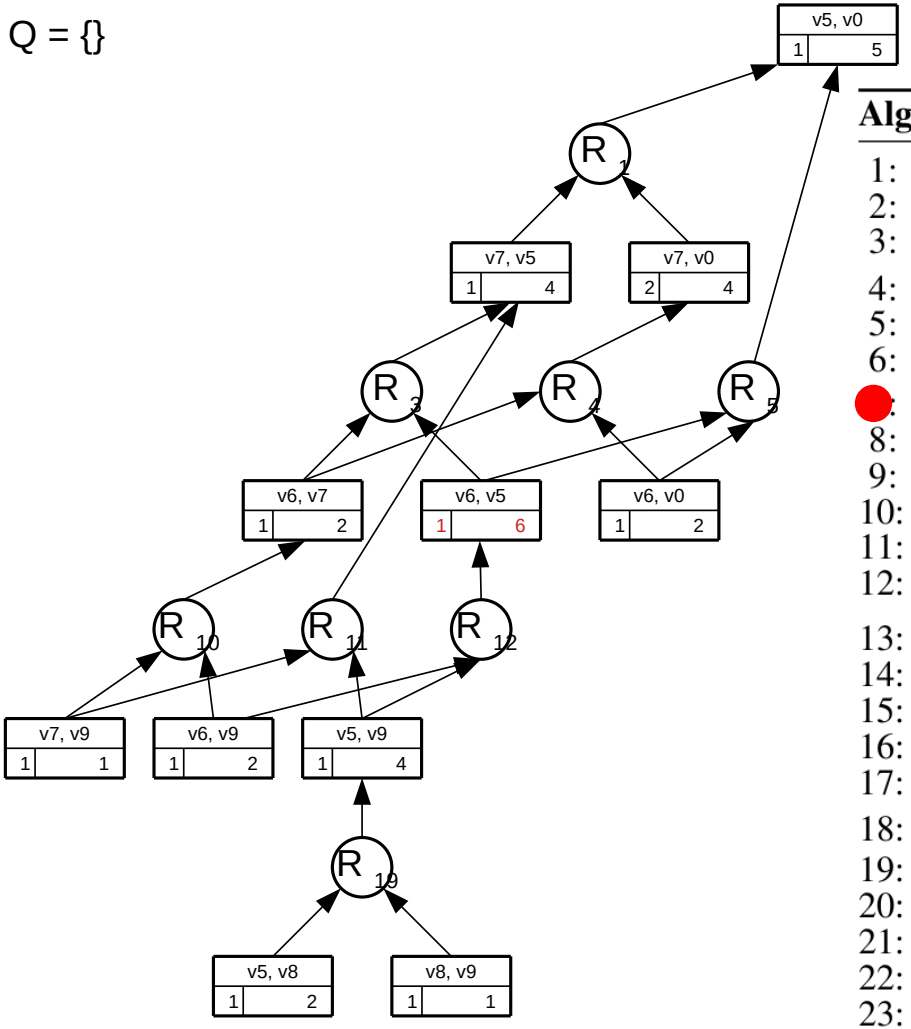
φ = 6

e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_r) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

Q = {}



e = (v6, v9); k = 2

Algorithm 4 DCH$_{scs}$-WInc (G^*, G, e, k)

```
1:  PriorityQueue Q ← ∅;
2:  if φ(v_e, G*) = φ(e, G) then
3:      c_φ(v_e) ← c_φ(v_e) − 1;
4:  φ(e, G) ← k;
5:  if c_φ(v_e) < 1 then
6:      Q.push(v_e);
7:  while Q ≠ ∅ do
8:      v_s ← Q.pop(); updateWeight(G*, v'_s);
9:      for each v_r ∈ nbr+(v_s) do
10:         v'_s ← nbr+(v_r);
11:         if c_φ(v'_s) < 1 then
12:             if v'_s ∉ Q then Q.push(v'_s);
13: procedure updateWeight(G*, v_s)
14:     for each v_r ∈ nbr+(v_s) do
15:         v_s1, v_s2 ← nbr−(v_r); v_s3 ← nbr+(v_r);
16:         if φ(v_s3, G*) = φ(v_s1, G*) + φ(v_s2, G*) then
17:             c_φ(v_s3) ← c_φ(v_s3) − 1;
18:     φ ← φ(s, G); c_φ(v_s) ← 1;
19:     for each v_r ∈ nbr−(v_s) do
20:         v_s1, v_s2 ← nbr−(v_r);
21:         if φ > φ(v_s1, G*) + φ(v_s2, G*) then
22:             φ ← φ(v_s1, G*) + φ(v_s2, G*); c_φ(v_s) ← 1;
23:         else if φ = φ(v_s1, G*) + φ(v_s2, G*) then
24:             c_φ(v_s) ← c_φ(v_s) + 1;
25:     φ(v_s, G*) ← φ;
```

φ = 6