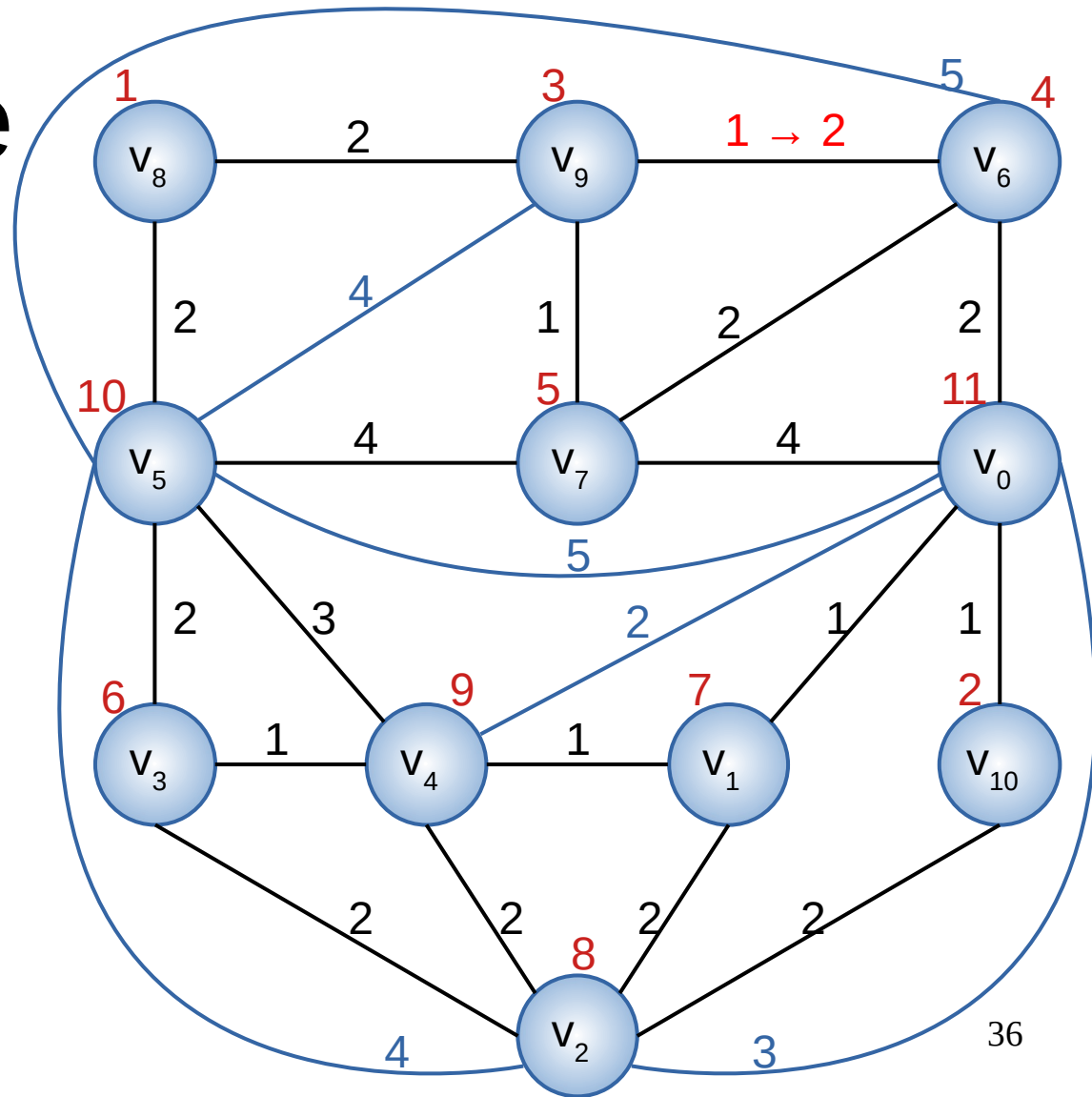
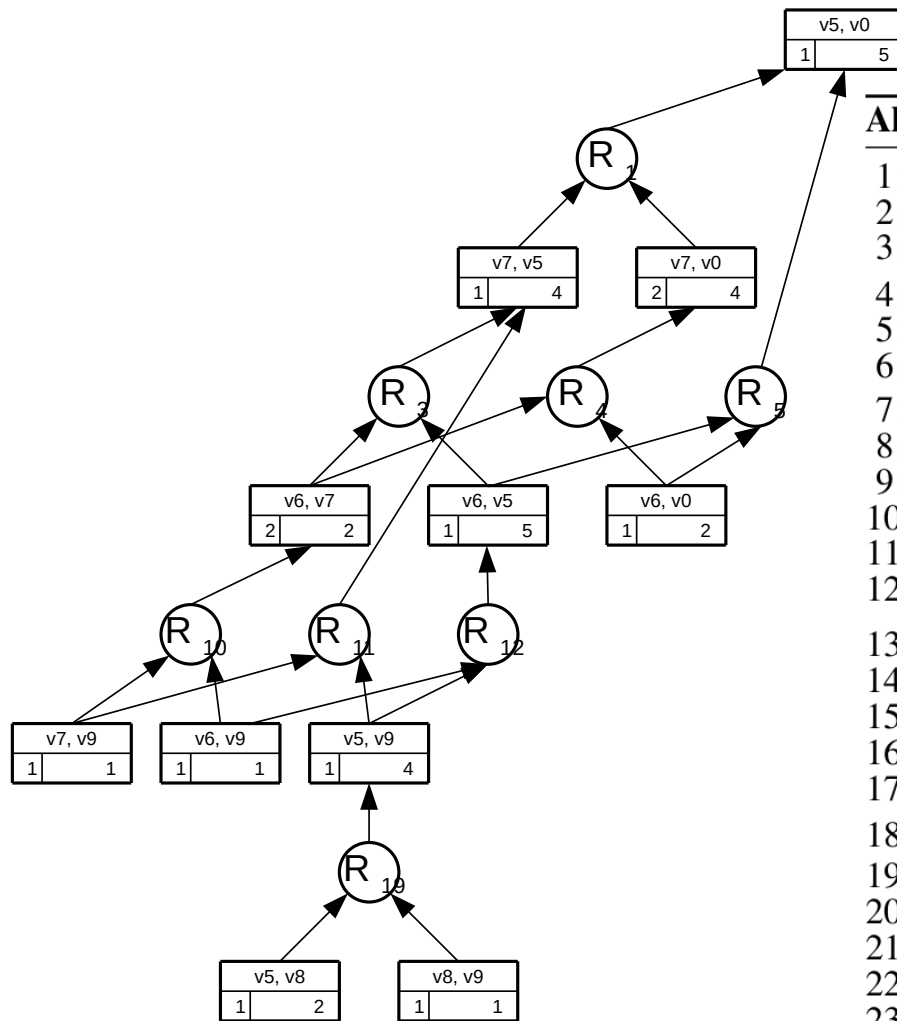


Weight Increase

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





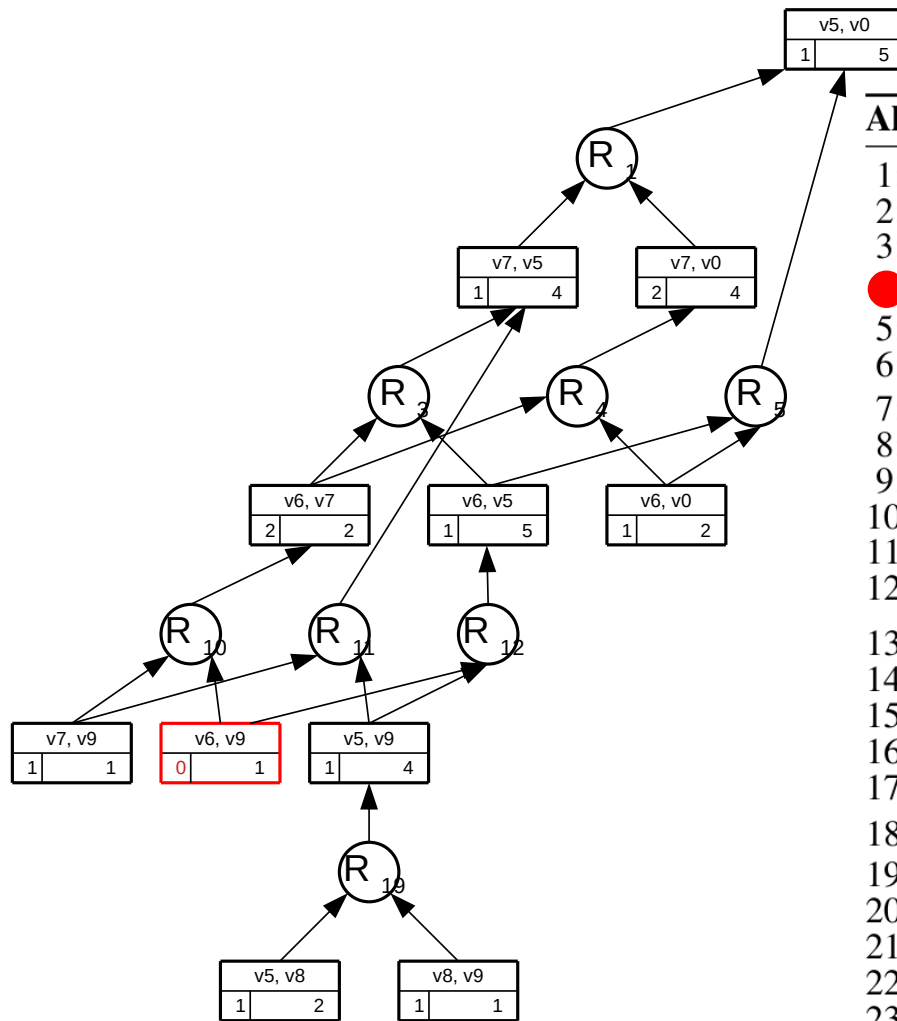
$e = (v6, v9); k = 2$

Algorithm 4 $DCH_{SCS-WInc}(G^*, G, e, k)$

```

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```



$e = (v6, v9); k = 2$

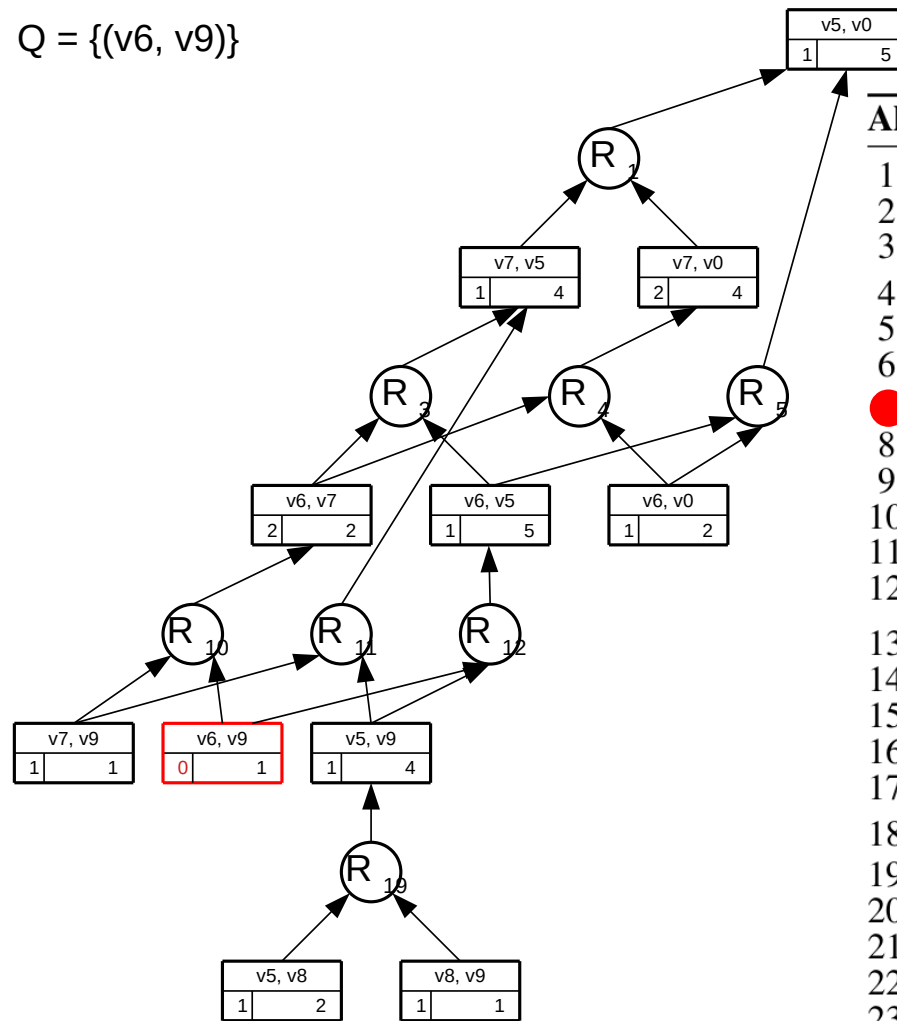
Algorithm 4 DCH_{SCS}-WInc (G^*, G, e, k)

```

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  $v_r \in nbr^+(v_s)$  do
10:       $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:        $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:       if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:          $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:        $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:       for each  $v_r \in nbr^-(v_s)$  do
20:          $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:         if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:            $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:         else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:            $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:        $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{(v6, v9)\}$



$e = (v6, v9); k = 2$

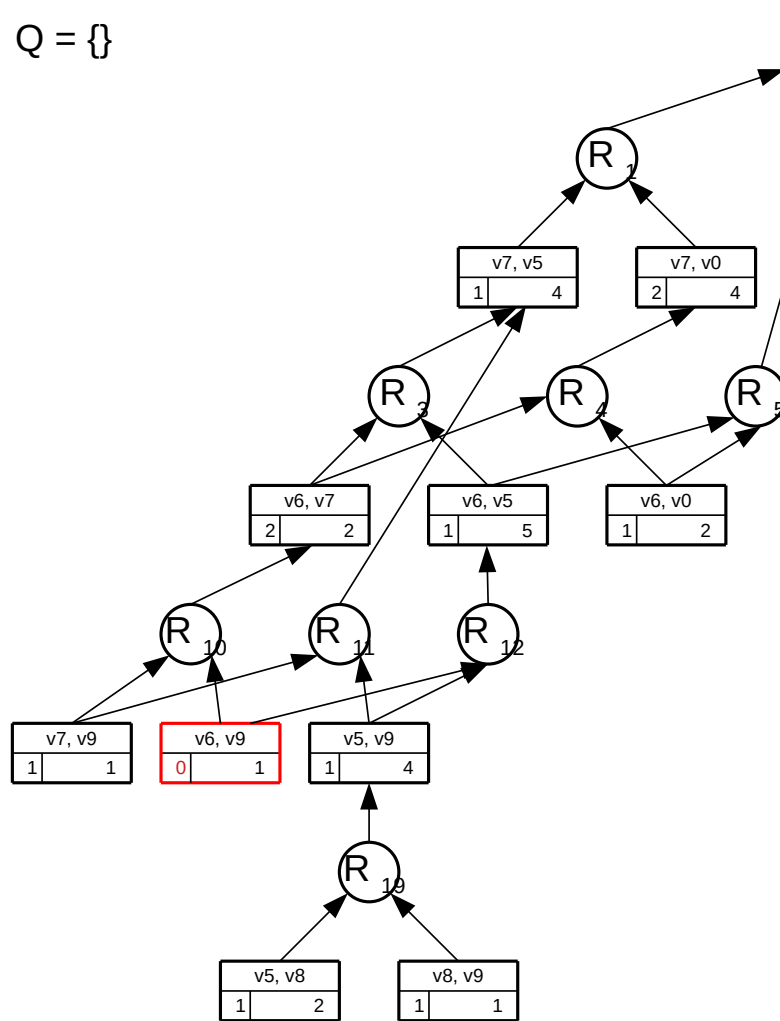
Algorithm 4 $DCH_{SCS}\text{-}WInc(G^*, G, e, k)$

```

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  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 $DCH_{SCS}\text{-}WInc(G^*, G, e, k)$

```

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

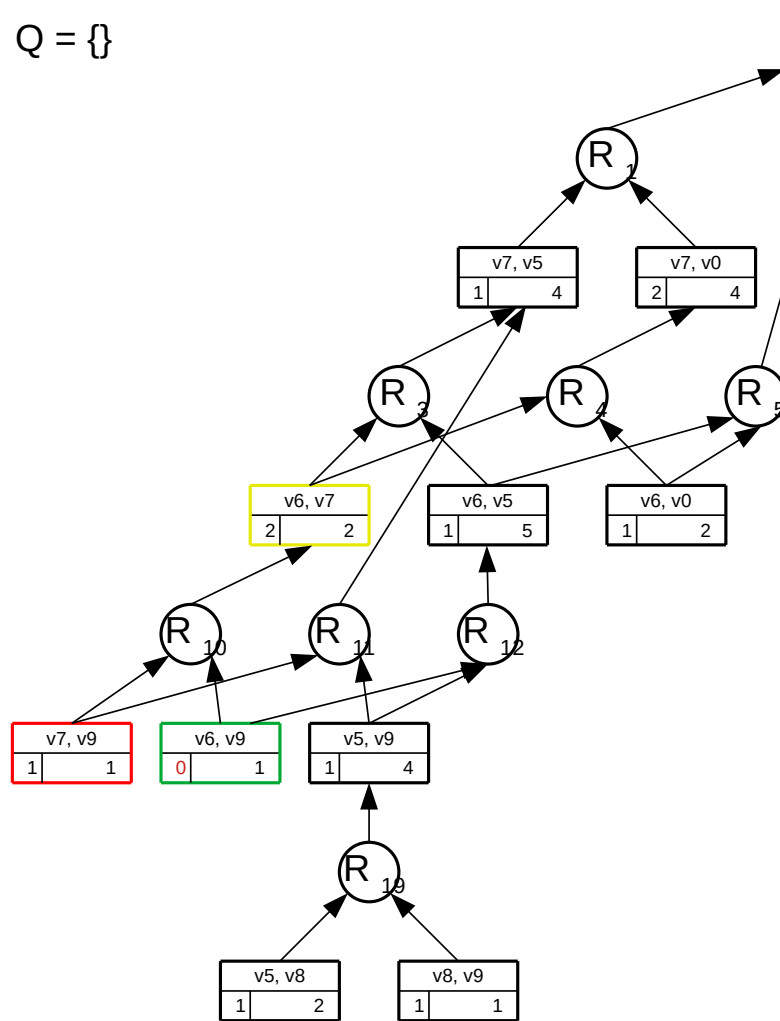
[illegible]
$$e = (v_6, v_9); k = 2$$
Algorithm 4 $\text{DCH}_{\text{scs-WInc}}(G^*, G, e, k)$

```

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:    $Q.\text{push}(v_e)$ ;
7: while  $Q \neq \emptyset$  do
8:    $v_s \leftarrow Q.\text{pop}()$ ;  $\text{updateWeight}(G^*, v'_s)$ ;
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  $Q.\text{push}(v'_s)$ ;
13: procedure  $\text{updateWeight}(G^*, v_s)$ 
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:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{\}$



$e = (v6, v9); k = 2$

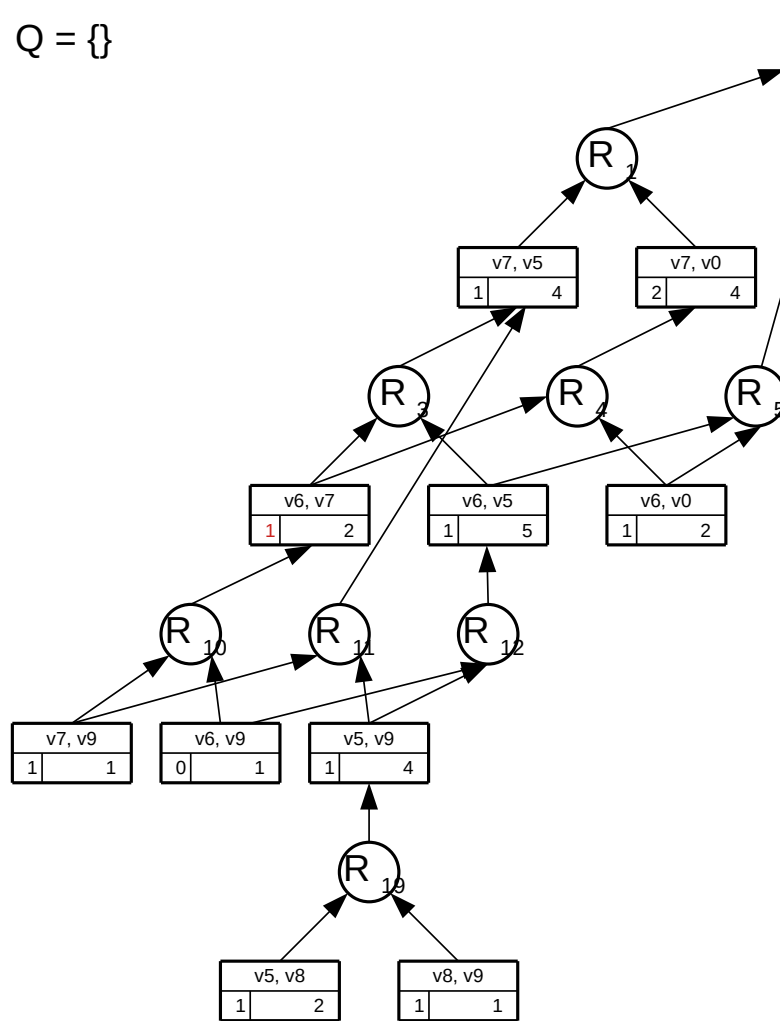
Algorithm 4 $DCH_{SCS}\text{-}WInc(G^*, G, e, k)$

```

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  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 $DCH_{SCS}\text{-}WInc(G^*, G, e, k)$

```

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:     $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:    for each  $v_r \in nbr^-(v_s)$  do
20:       $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:      if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:         $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:      else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:         $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:     $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{\}$

Diagram illustrating a search space for a 3-disk Tower of Hanoi problem. The nodes are labeled R_0 through R_{12} . The nodes are represented by rectangles (states) and circles (goals). The states are represented by rectangles with a 2x2 table structure:

- R_0 : $\begin{bmatrix} v7, v9 \\ 1 & 1 \end{bmatrix}$
- R_1 : $\begin{bmatrix} v6, v9 \\ 0 & 1 \end{bmatrix}$
- R_2 : $\begin{bmatrix} v5, v9 \\ 1 & 4 \end{bmatrix}$
- R_3 : $\begin{bmatrix} v6, v7 \\ 1 & 2 \end{bmatrix}$
- R_4 : $\begin{bmatrix} v6, v5 \\ 1 & 5 \end{bmatrix}$
- R_5 : $\begin{bmatrix} v6, v0 \\ 1 & 2 \end{bmatrix}$
- R_6 : $\begin{bmatrix} v7, v5 \\ 1 & 4 \end{bmatrix}$
- R_7 : $\begin{bmatrix} v7, v0 \\ 2 & 4 \end{bmatrix}$
- R_8 : $\begin{bmatrix} v5, v0 \\ 1 & 5 \end{bmatrix}$
- R_9 : $\begin{bmatrix} v5, v8 \\ 1 & 2 \end{bmatrix}$
- R_{10} : $\begin{bmatrix} v8, v9 \\ 1 & 1 \end{bmatrix}$
- R_{11} : $\begin{bmatrix} v7, v9 \\ 1 & 1 \end{bmatrix}$
- R_{12} : $\begin{bmatrix} v6, v9 \\ 0 & 1 \end{bmatrix}$

The nodes are connected by directed edges, forming a search tree. The goal node is R_{11} , which is highlighted in red. The search space is defined by the sequence of states $R_0, R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{12}$.

$$e = (v_6, v_9); k = 2$$
Algorithm 4 $\text{DCH}_{\text{scs-WInc}}(G^*, G, e, k)$

```

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:    $Q.\text{push}(v_e)$ ;
7: while  $Q \neq \emptyset$  do
8:    $v_s \leftarrow Q.\text{pop}()$ ;  $\text{updateWeight}(G^*, v'_s)$ ;
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  $Q.\text{push}(v'_s)$ ;
13: procedure  $\text{updateWeight}(G^*, v_s)$ 
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:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

The graph illustrates a complex dependency structure. The nodes are arranged in a hierarchical manner, with R_{19} at the bottom and R_0 at the top. The nodes are represented by rectangles (variables) and circles (relations). The edges represent dependencies between these nodes. A red circle highlights node R_{17} . A red box highlights node v_7, v_9 . A green box highlights node v_5, v_9 . A yellow box highlights node v_7, v_5 .

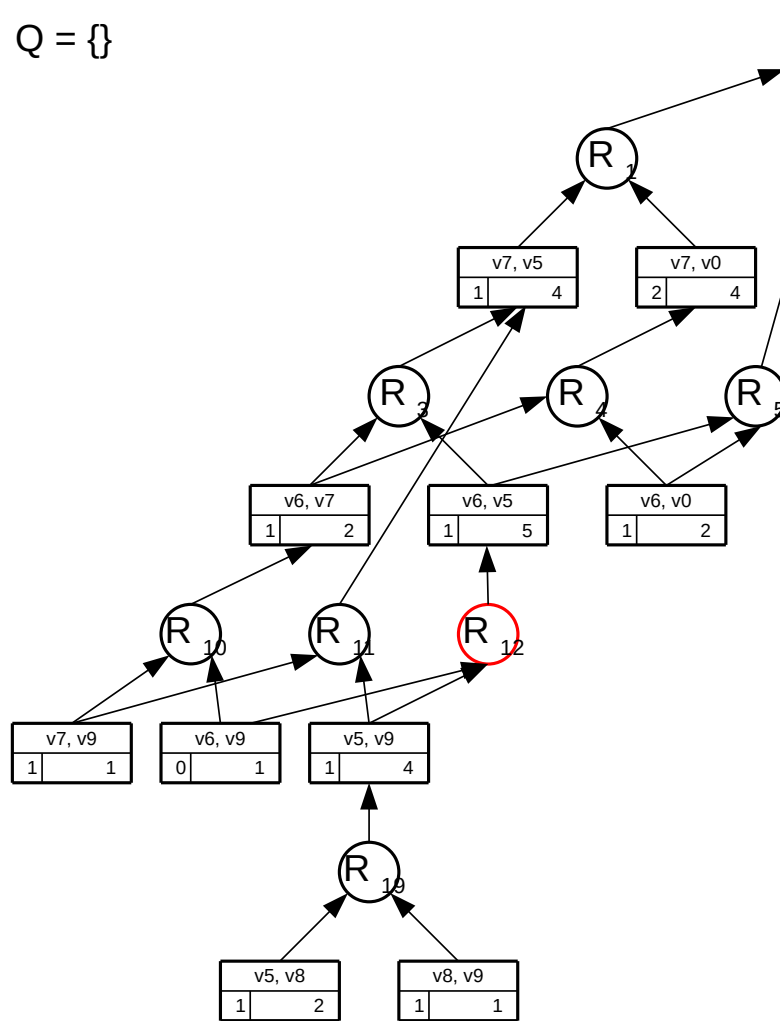
$$e = (v_6, v_9); k = 2$$
Algorithm 4 $\text{DCH}_{\text{SCS-WInc}}(G^*, G, e, k)$

```

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:    $Q.\text{push}(v_e)$ ;
7: while  $Q \neq \emptyset$  do
8:    $v_s \leftarrow Q.\text{pop}()$ ;  $\text{updateWeight}(G^*, v'_s)$ ;
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  $Q.\text{push}(v'_s)$ ;
13: procedure  $\text{updateWeight}(G^*, v_s)$ 
14:   for each  $v_r \in \text{nbr}^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow \text{nbr}^-(v_r)$ ;  $v_{s3} \leftarrow \text{nbr}^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in \text{nbr}^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow \text{nbr}^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  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:    $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$ ;
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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r); v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G); c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*); c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

The graph consists of nodes R_0 through R_{12} and v_0 through v_9 . Each node is a table with two rows. The top row contains a pair of vertices, and the bottom row contains two numbers. Edges connect nodes in a hierarchical structure. A red circle highlights node R_{10} .

Node details:

- R_0 : v_5, v_0 (1, 5)
- R_1 : v_7, v_5 (1, 4)
- R_2 : v_7, v_0 (2, 4)
- R_3 : v_6, v_7 (1, 2)
- R_4 : v_6, v_5 (1, 5)
- R_5 : v_6, v_0 (1, 2)
- R_6 : v_7, v_9 (1, 1)
- R_7 : v_6, v_9 (0, 1)
- R_8 : v_5, v_9 (1, 4)
- R_9 : v_5, v_8 (1, 2)
- R_{10} : v_8, v_9 (1, 1)
- R_{11} : v_6, v_7 (1, 2)
- R_{12} : v_6, v_5 (1, 5)

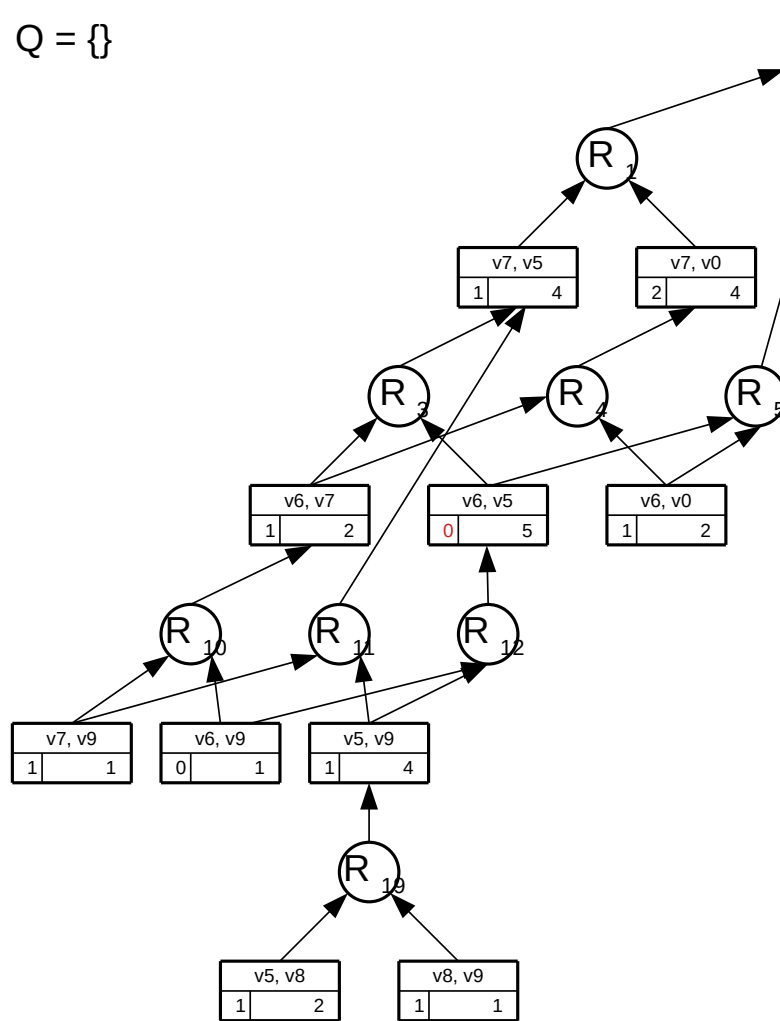
Edges (from parent to child):

- $R_0 \rightarrow R_1$
- $R_0 \rightarrow R_2$
- $R_1 \rightarrow R_3$
- $R_1 \rightarrow R_4$
- $R_2 \rightarrow R_4$
- $R_2 \rightarrow R_5$
- $R_3 \rightarrow R_6$
- $R_3 \rightarrow R_7$
- $R_4 \rightarrow R_8$
- $R_5 \rightarrow R_8$
- $R_6 \rightarrow R_9$
- $R_7 \rightarrow R_9$
- $R_8 \rightarrow R_{10}$
- $R_9 \rightarrow R_{10}$
- $R_{10} \rightarrow R_{11}$
- $R_{10} \rightarrow R_{12}$

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

47

$Q = \{\}$



$e = (v6, v9); k = 2$

Algorithm 4 $DCH_{SCS}\text{-}WInc(G^*, G, e, k)$

```

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r); v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G); c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{ \}$

$\phi = 2$

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

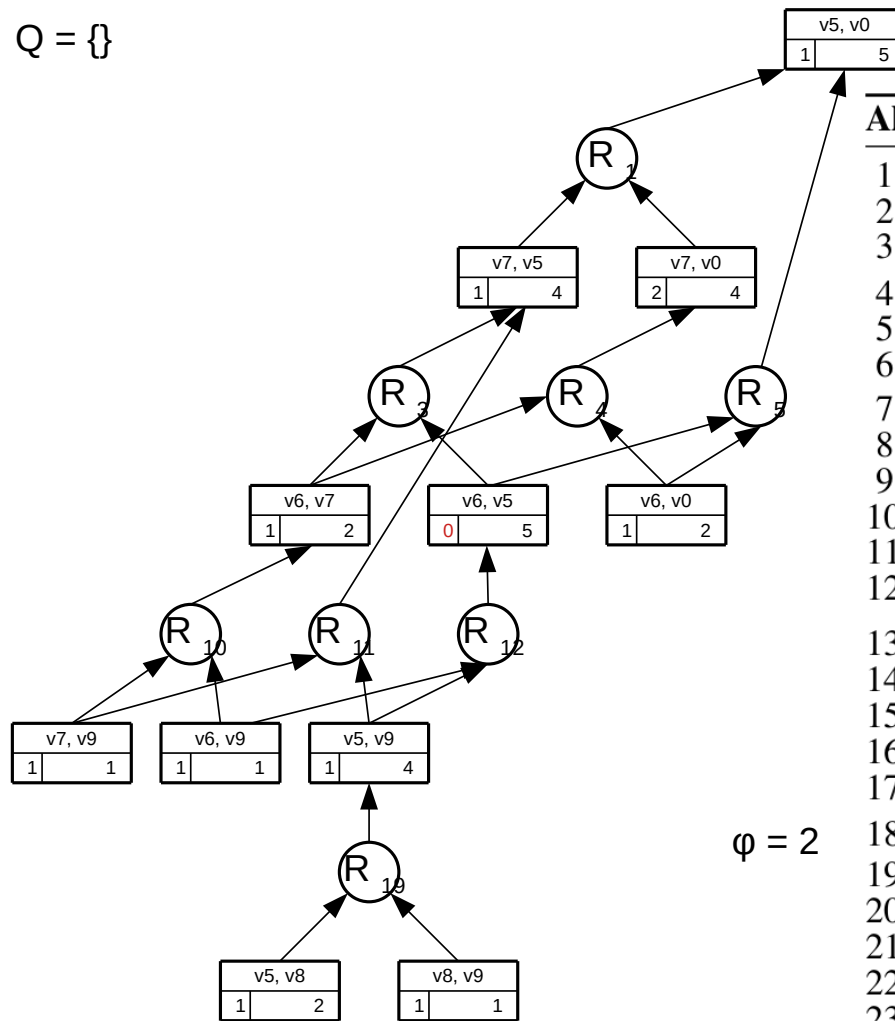
```

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:    $Q.\text{push}(v_e)$ ;
7: while  $Q \neq \emptyset$  do
8:    $v_s \leftarrow Q.\text{pop}()$ ;  $\text{updateWeight}(G^*, v'_s)$ ;
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  $Q.\text{push}(v'_s)$ ;
13: procedure  $\text{updateWeight}(G^*, v_s)$ 
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:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

[illegible]**Algorithm 4** $\text{DCH}_{\text{SCS-WInc}}(G^*, G, e, k)$ 

$Q = \{\}$



$e = (v6, v9); k = 2$

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do  $nbr^-(v_s) = \{\}$ 
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```


The diagram illustrates a sequence of nodes R_0 through R_{12} and v_0 through v_9 . Each node is a rectangle with two rows. The top row contains vertex labels, and the bottom row contains two numbers. Some numbers are red.

- R_0 : Top row: v_5, v_0 ; Bottom row: 1, 5
- R_1 : Top row: v_7, v_5 ; Bottom row: 1, 4
- R_2 : Top row: v_7, v_0 ; Bottom row: 2, 4
- R_3 : Top row: v_6, v_7 ; Bottom row: 1, 2
- R_4 : Top row: v_6, v_5 ; Bottom row: 0, 5 (0 is red)
- R_5 : Top row: v_6, v_0 ; Bottom row: 1, 2
- R_6 : Top row: v_7, v_9 ; Bottom row: 1, 1
- R_7 : Top row: v_6, v_9 ; Bottom row: 1, 2 (2 is red)
- R_8 : Top row: v_5, v_9 ; Bottom row: 1, 4
- R_9 : Top row: v_5, v_8 ; Bottom row: 1, 2
- R_{10} : Top row: v_8, v_9 ; Bottom row: 1, 1
- R_{11} : Top row: v_7, v_5 ; Bottom row: 1, 4
- R_{12} : Top row: v_7, v_0 ; Bottom row: 2, 4

Arrows indicate directed edges between nodes. A red dot is next to R_8 .

$$e = (v_6, v_9); k = 2$$
Algorithm 4 $\text{DCH}_{\text{scs-WInc}}(G^*, G, e, k)$

```

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:    $Q.\text{push}(v_e)$ ;
7: while  $Q \neq \emptyset$  do
8:    $v_s \leftarrow Q.\text{pop}()$ ;  $\text{updateWeight}(G^*, v'_s)$ ;
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  $Q.\text{push}(v'_s)$ ;
13: procedure  $\text{updateWeight}(G^*, v_s)$ 
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:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

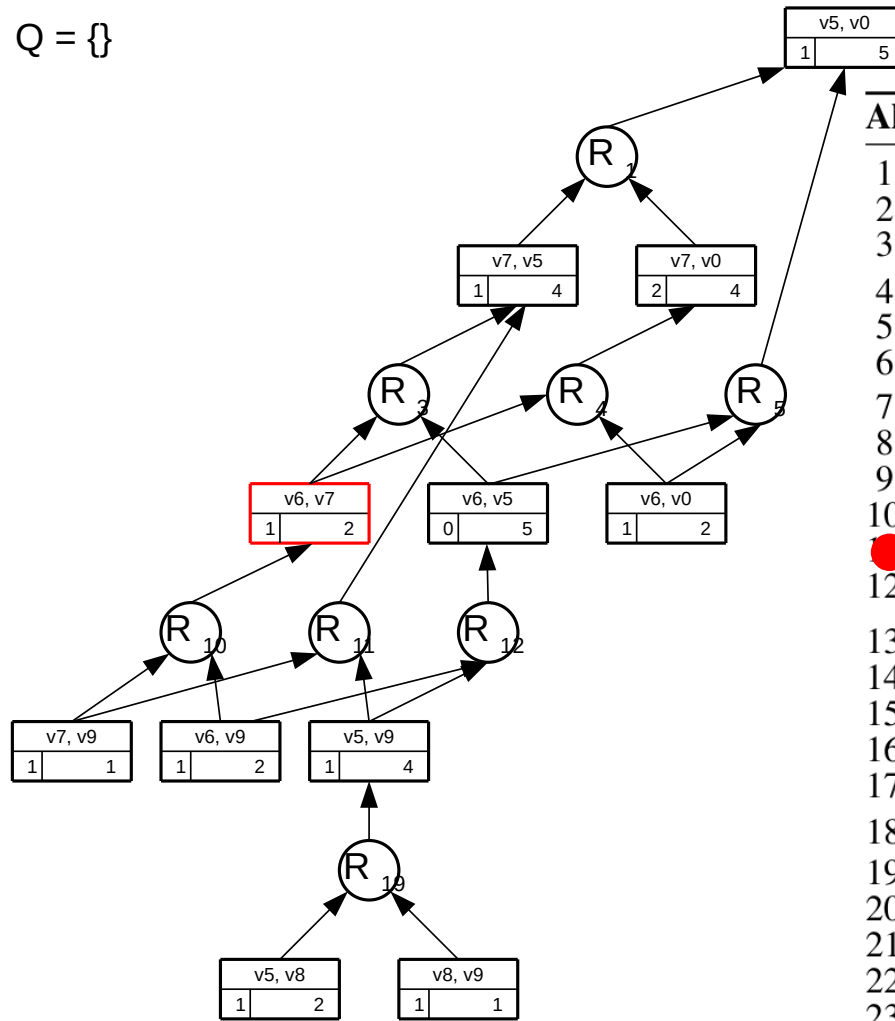
[illegible]
$$e = (v_6, v_9); k = 2$$
Algorithm 4 $\text{DCH}_{\text{scs-WInc}}(G^*, G, e, k)$

```

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:    $Q.\text{push}(v_e)$ ;
7: while  $Q \neq \emptyset$  do
8:    $v_s \leftarrow Q.\text{pop}()$ ;  $\text{updateWeight}(G^*, v'_s)$ ;
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  $Q.\text{push}(v'_s)$ ;
13: procedure  $\text{updateWeight}(G^*, v_s)$ 
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:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{\}$



$e = (v6, v9); k = 2$

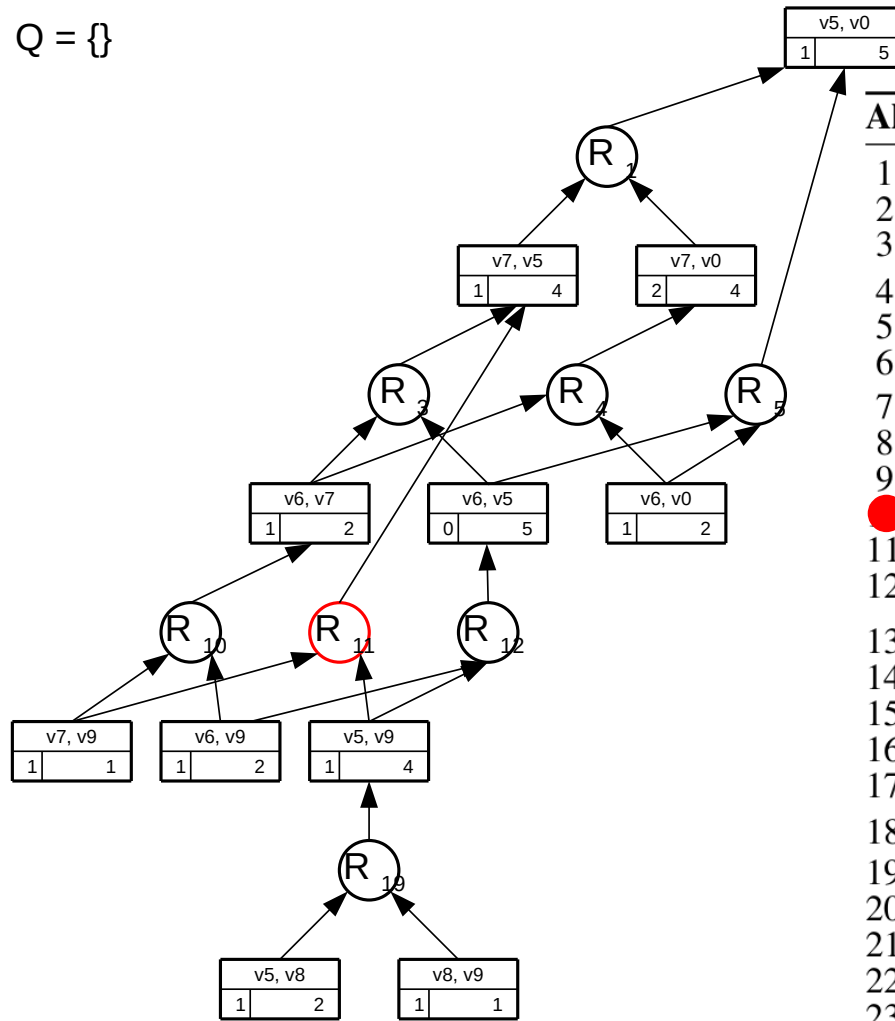
Algorithm 4 DCH_{scs}-WInc (G^*, G, e, k)

```

1: PriorityQueue  $Q \leftarrow \emptyset$ ;
2: if  $\phi(v_e, G^*) = \phi(e, G)$  then
3:    $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$ ;
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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:      $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:     if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:        $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:      $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:     if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:        $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:     else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:        $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{\}$



$e = (v_6, v_9); k = 2$

Algorithm 4 $DCH_{SCS}\text{-}WInc(G^*, G, e, k)$

```

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

Q = {}

Diagram illustrating a search space for a 3-disk Tower of Hanoi problem. The graph shows states (nodes) and transitions (edges). Each state is represented by a table with two rows and two columns, indicating the disks on the start and goal pegs.

States (Nodes):

- R_1 :

v5, v8
1 2
- R_2 :

v8, v9
1 1
- R_3 :

v5, v9
1 4
- R_4 :

v6, v9
1 2
- R_5 :

v7, v9
1 1
- R_6 :

v6, v7
1 2
- R_7 :

v6, v5
0 5
- R_8 :

v7, v5
1 4

 (Goal State)
- R_9 :

v7, v0
2 4
- R_{10} :

v5, v0
1 5

Transitions (Edges):

- $R_1 \rightarrow R_2$
- $R_1 \rightarrow R_3$
- $R_2 \rightarrow R_4$
- $R_3 \rightarrow R_5$
- $R_4 \rightarrow R_6$
- $R_5 \rightarrow R_7$
- $R_6 \rightarrow R_8$
- $R_7 \rightarrow R_9$
- $R_8 \rightarrow R_{10}$
- $R_9 \rightarrow R_{10}$

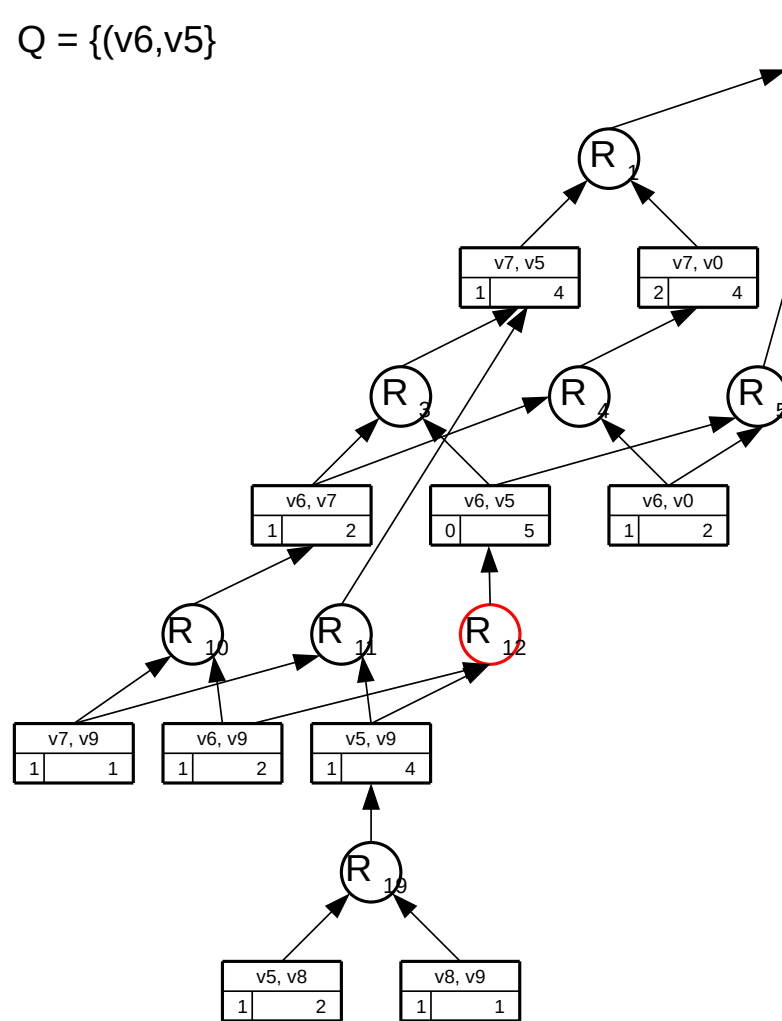
$$e = (v_6, v_9); k = 2$$
Algorithm 4 $\text{DCH}_{\text{SCS-WInc}}(G^*, G, e, k)$

```

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:    $Q.\text{push}(v_e)$ ;
7: while  $Q \neq \emptyset$  do
8:    $v_s \leftarrow Q.\text{pop}()$ ;  $\text{updateWeight}(G^*, v'_s)$ ;
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  $Q.\text{push}(v'_s)$ ;
13: procedure  $\text{updateWeight}(G^*, v_s)$ 
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:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{(v6, v5)\}$



$e = (v6, v9); k = 2$

Algorithm 4 $DCH_{SCS}\text{-}WInc(G^*, G, e, k)$

```

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

[illegible]
$$e = (v_6, v_9); k = 2$$
Algorithm 4 $\text{DCH}_{\text{scs-WInc}}(G^*, G, e, k)$

```

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:    $Q.\text{push}(v_e)$ ;
7: while  $Q \neq \emptyset$  do
8:    $v_s \leftarrow Q.\text{pop}()$ ;  $\text{updateWeight}(G^*, v'_s)$ ;
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  $Q.\text{push}(v'_s)$ ;
13: procedure  $\text{updateWeight}(G^*, v_s)$ 
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:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

The diagram illustrates a sequence of nodes R_0 through R_{19} and their associated data blocks. Each node R_i is represented by a circle, and each data block is a rectangle containing a pair of vertices and a 2x2 matrix. The blocks are connected by arrows indicating a sequence or relationship.

The data blocks are as follows:

- R_0 : v_5, v_0 (1 | 5)
- R_1 : v_7, v_5 (1 | 4)
- R_2 : v_7, v_0 (2 | 4)
- R_3 : v_6, v_7 (1 | 2)
- R_4 : v_6, v_5 (0 | 5) (highlighted with a red border)
- R_5 : v_6, v_0 (1 | 2)
- R_6 : v_7, v_9 (1 | 1)
- R_7 : v_6, v_9 (1 | 2)
- R_8 : v_5, v_9 (1 | 4)
- R_9 : v_5, v_8 (1 | 2)
- R_{10} : v_8, v_9 (1 | 1)
- R_{11} : v_6, v_7 (1 | 2)
- R_{12} : v_6, v_5 (0 | 5) (highlighted with a red border)
- R_{13} : v_7, v_5 (1 | 4)
- R_{14} : v_7, v_0 (2 | 4)
- R_{15} : v_6, v_7 (1 | 2)
- R_{16} : v_6, v_5 (0 | 5) (highlighted with a red border)
- R_{17} : v_6, v_0 (1 | 2)
- R_{18} : v_7, v_9 (1 | 1)
- R_{19} : v_5, v_8 (1 | 2)

The sequence of nodes is as follows:

- R_{19} is the root node.
- R_{19} branches into R_{18} and R_{17} .
- R_{18} leads to R_{16} .
- R_{16} branches into R_{15} and R_{14} .
- R_{15} leads to R_{13} .
- R_{13} branches into R_{12} and R_{11} .
- R_{12} leads to R_{10} .
- R_{10} branches into R_9 and R_8 .
- R_9 leads to R_7 .
- R_7 branches into R_6 and R_5 .
- R_6 leads to R_4 .
- R_4 branches into R_3 and R_2 .
- R_3 leads to R_1 .
- R_1 branches into R_0 and R_{19} .

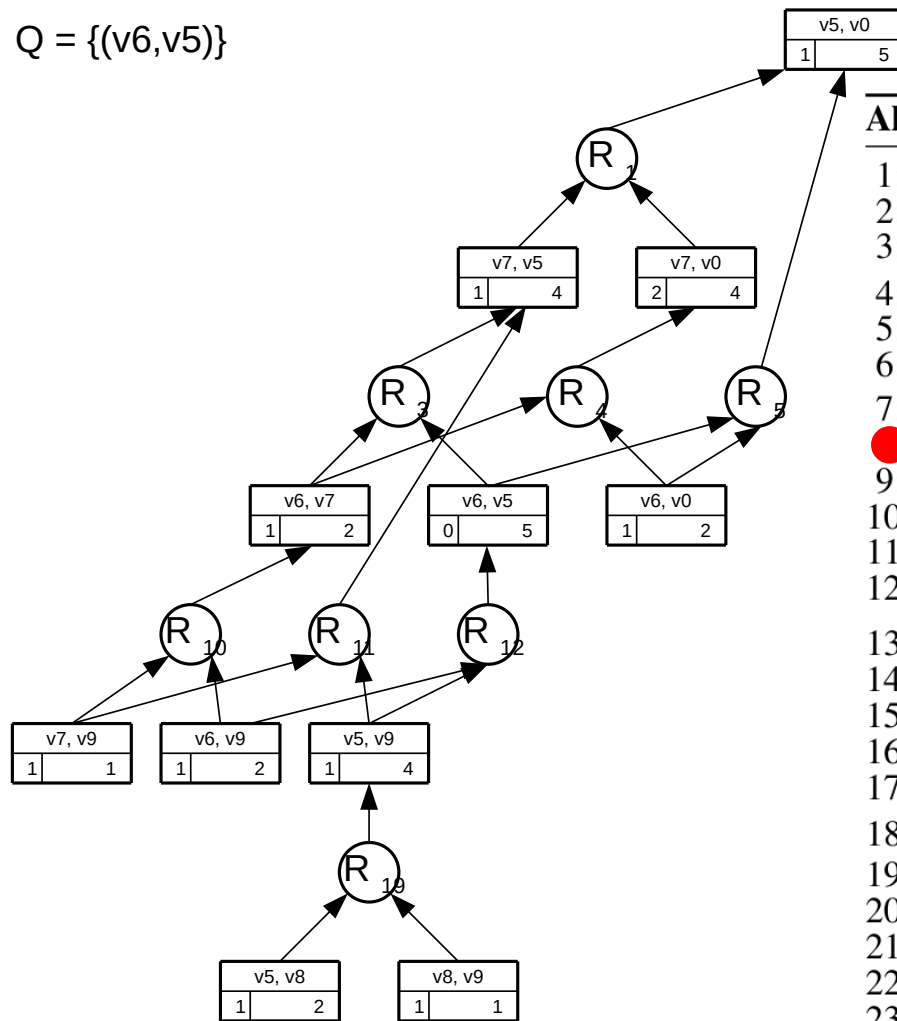
$$e = (v_6, v_9); k = 2$$
Algorithm 4 $\text{DCH}_{\text{scs-WInc}}(G^*, G, e, k)$

```

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:    $Q.\text{push}(v_e)$ ;
7: while  $Q \neq \emptyset$  do
8:    $v_s \leftarrow Q.\text{pop}()$ ;  $\text{updateWeight}(G^*, v'_s)$ ;
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  $Q.\text{push}(v'_s)$ ;
13: procedure  $\text{updateWeight}(G^*, v_s)$ 
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:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```


$Q = \{(v6, v5)\}$



$e = (v6, v9); k = 2$

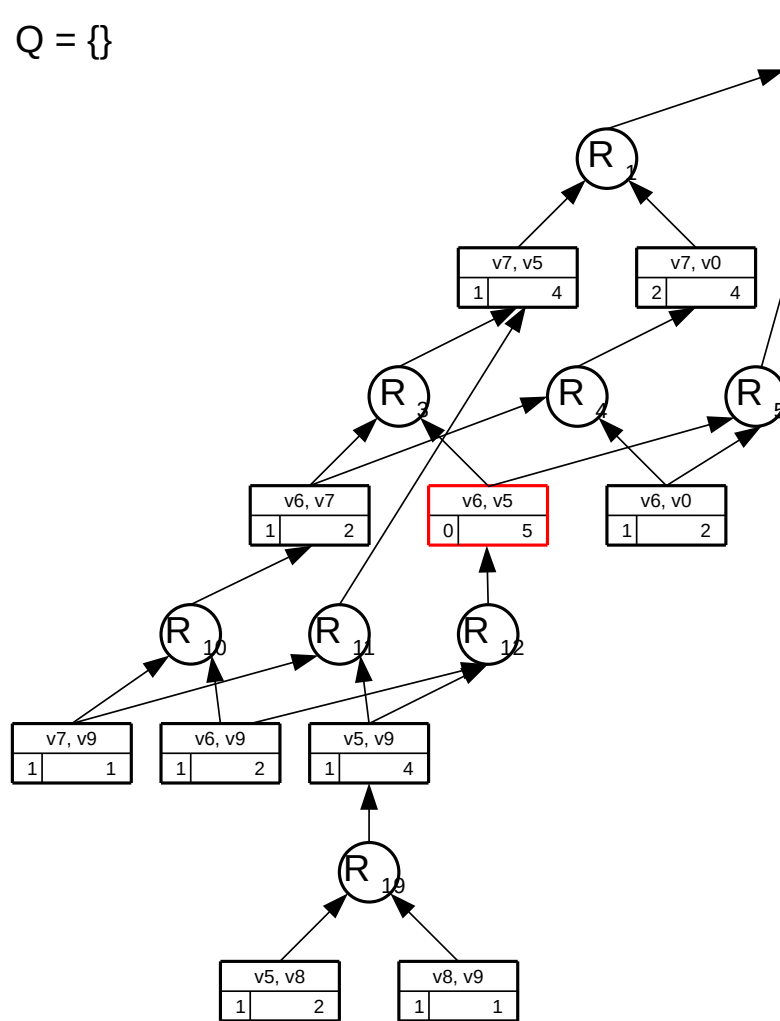
Algorithm 4 $DCH_{SCS}\text{-}WInc(G^*, G, e, k)$

```

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{\}$



$e = (v6, v9); k = 2$

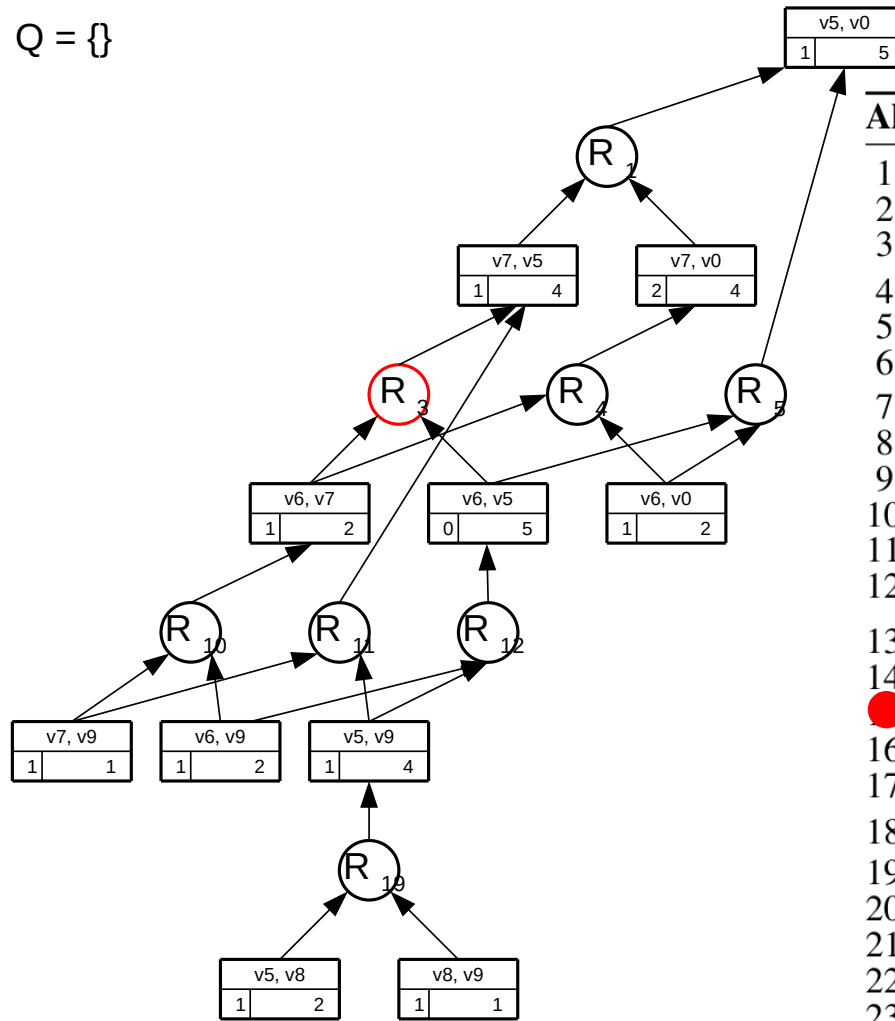
Algorithm 4 DCH_{scs}-WInc (G^*, G, e, k)

```

1: PriorityQueue  $Q \leftarrow \emptyset$ ;
2: if  $\phi(v_e, G^*) = \phi(e, G)$  then
3:    $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$ ;
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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{\}$



$e = (v6, v9); k = 2$

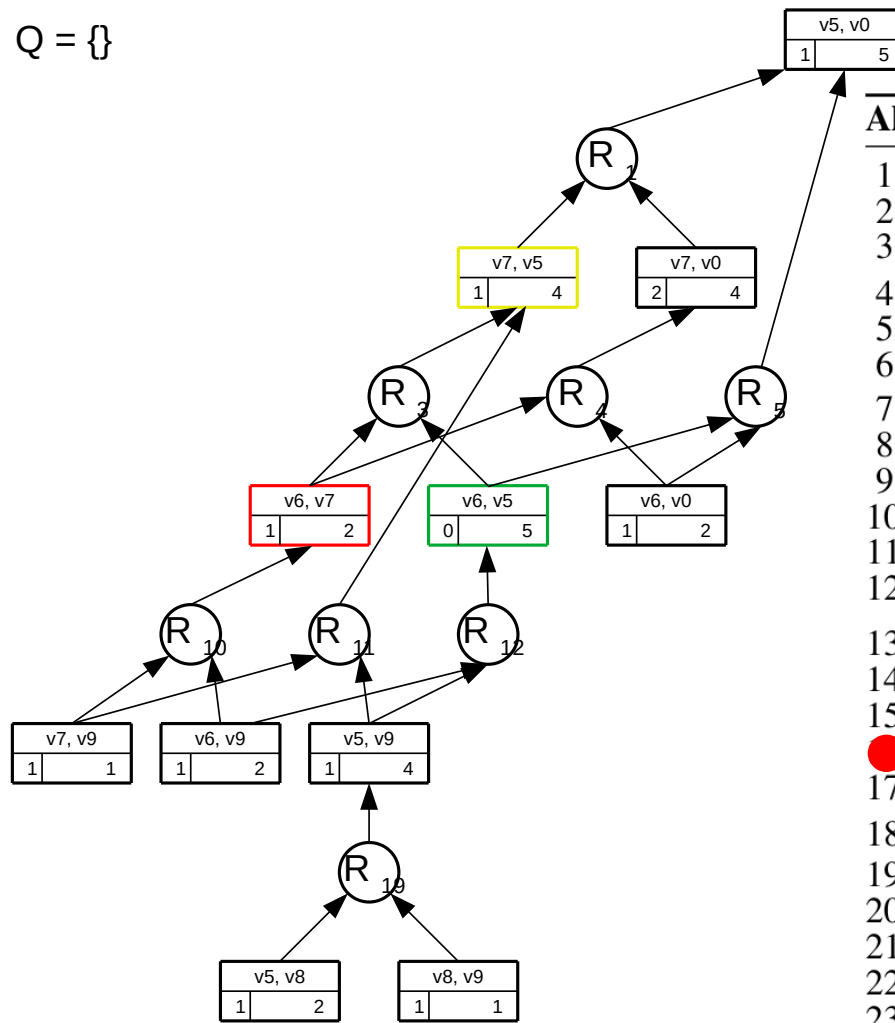
Algorithm 4 $DCH_{SCS-WInc}(G^*, G, e, k)$

```

1: PriorityQueue  $Q \leftarrow \emptyset$ ;
2: if  $\phi(v_e, G^*) = \phi(e, G)$  then
3:    $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$ ;
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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  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:    $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$ ;
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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:      $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:     if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:        $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:      $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:     if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:        $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:     else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:        $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

Diagram illustrating a sequence of nodes and their relationships:

- Nodes are represented as boxes with two rows: the top row contains a pair of vertex IDs, and the bottom row contains two numbers.
- Arrows indicate directed edges between nodes.
- Nodes R_5 and R_{10} are highlighted with red circles.
- A red dot is present next to the list of numbers on the right.

Node details:

- R_0 : v_5, v_0 (1, 5)
- R_1 : v_7, v_5 (1, 4)
- R_2 : v_7, v_0 (2, 4)
- R_3 : v_6, v_7 (1, 2)
- R_4 : v_6, v_5 (0, 5)
- R_5 : v_6, v_0 (1, 2)
- R_6 : v_7, v_9 (1, 1)
- R_7 : v_6, v_9 (1, 2)
- R_8 : v_5, v_9 (1, 4)
- R_9 : v_5, v_8 (1, 2)
- R_{10} : v_8, v_9 (1, 1)

Sequence of numbers on the right:

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23

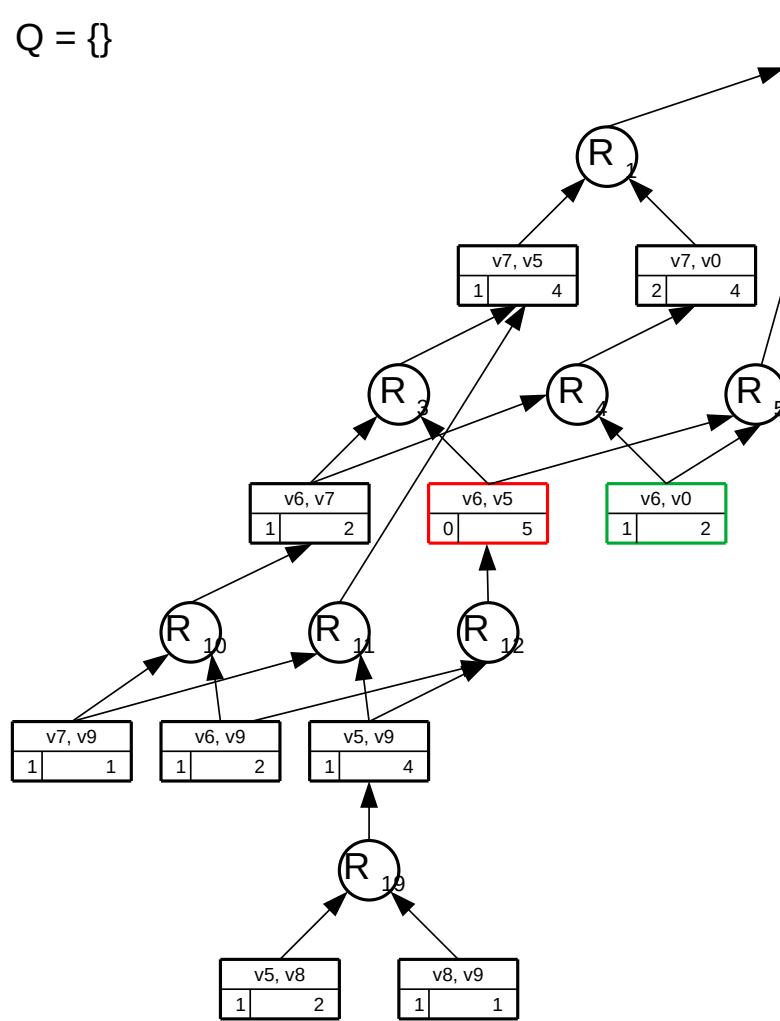
$$e = (v_6, v_9); k = 2$$
Algorithm 4 $\text{DCH}_{\text{SCS-WInc}}(G^*, G, e, k)$

```

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:    $Q.\text{push}(v_e)$ ;
7: while  $Q \neq \emptyset$  do
8:    $v_s \leftarrow Q.\text{pop}()$ ;  $\text{updateWeight}(G^*, v'_s)$ ;
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  $Q.\text{push}(v'_s)$ ;
13: procedure  $\text{updateWeight}(G^*, v_s)$ 
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:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{\}$



$e = (v6, v9); k = 2$

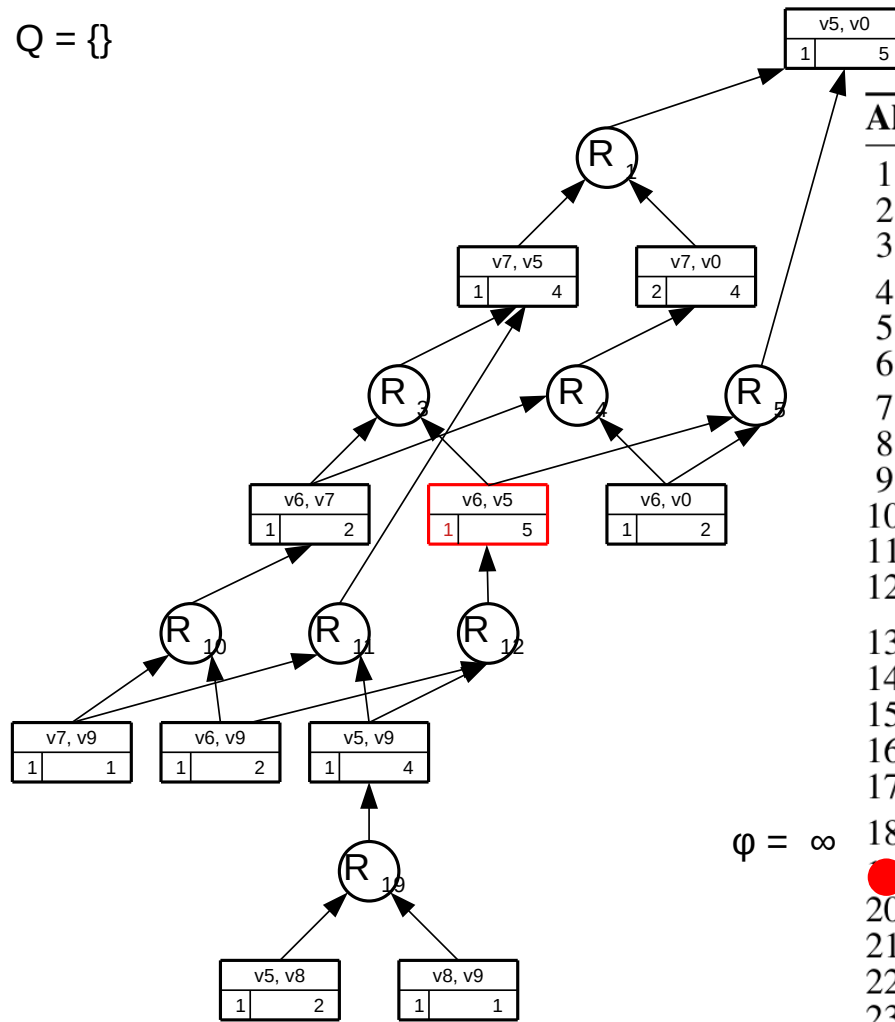
Algorithm 4 DCH_{scs}-WInc (G^*, G, e, k)

```

1: PriorityQueue  $Q \leftarrow \emptyset$ ;
2: if  $\phi(v_e, G^*) = \phi(e, G)$  then
3:    $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$ ;
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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:      $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:     if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:        $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:      $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:     if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:        $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:     else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:        $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{\}$



$e = (v6, v9); k = 2$

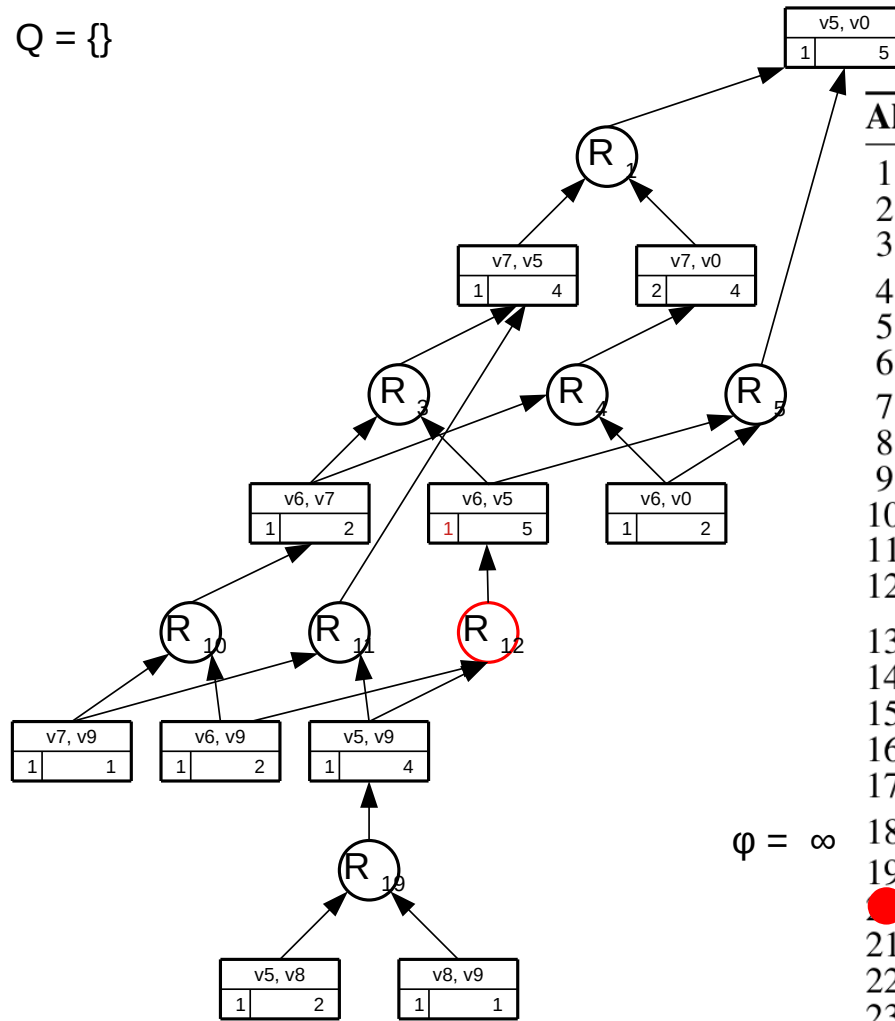
Algorithm 4 DCH_{scs}-WInc (G^*, G, e, k)

```

1: PriorityQueue  $Q \leftarrow \emptyset$ ;
2: if  $\phi(v_e, G^*) = \phi(e, G)$  then
3:    $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$ ;
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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{\}$



$e = (v_6, v_9); k = 2$

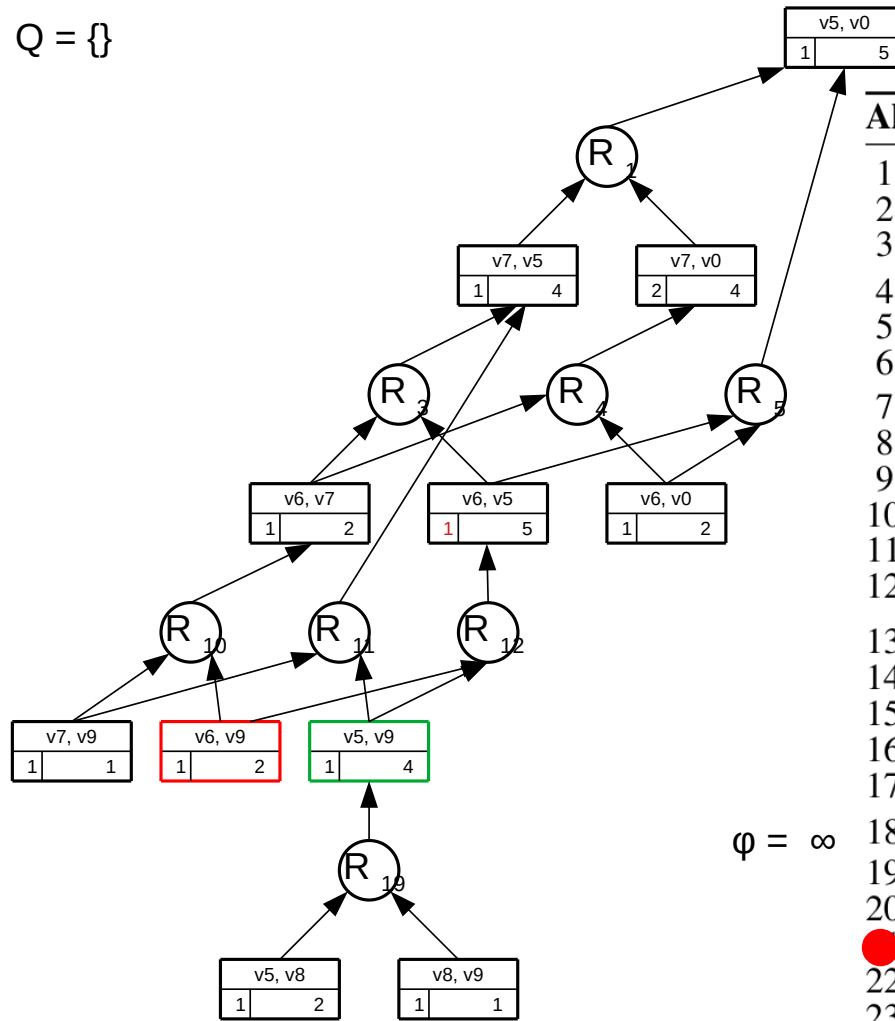
Algorithm 4 $DCH_{SCS-WInc}(G^*, G, e, k)$

```

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```


$Q = \{\}$



$e = (v6, v9); k = 2$

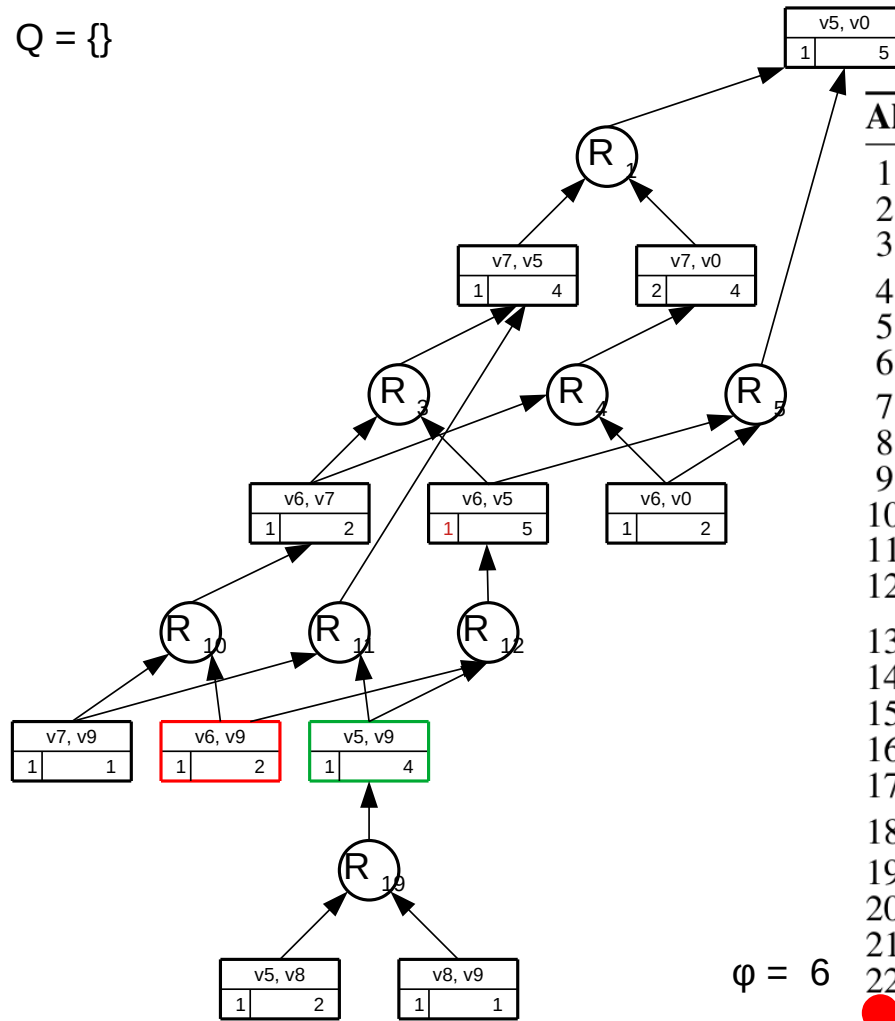
Algorithm 4 DCH_{scs}-WInc (G^*, G, e, k)

```

1: PriorityQueue  $Q \leftarrow \emptyset$ ;
2: if  $\phi(v_e, G^*) = \phi(e, G)$  then
3:    $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$ ;
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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$Q = \{\}$



$\phi = 6$

$e = (v6, v9); k = 2$

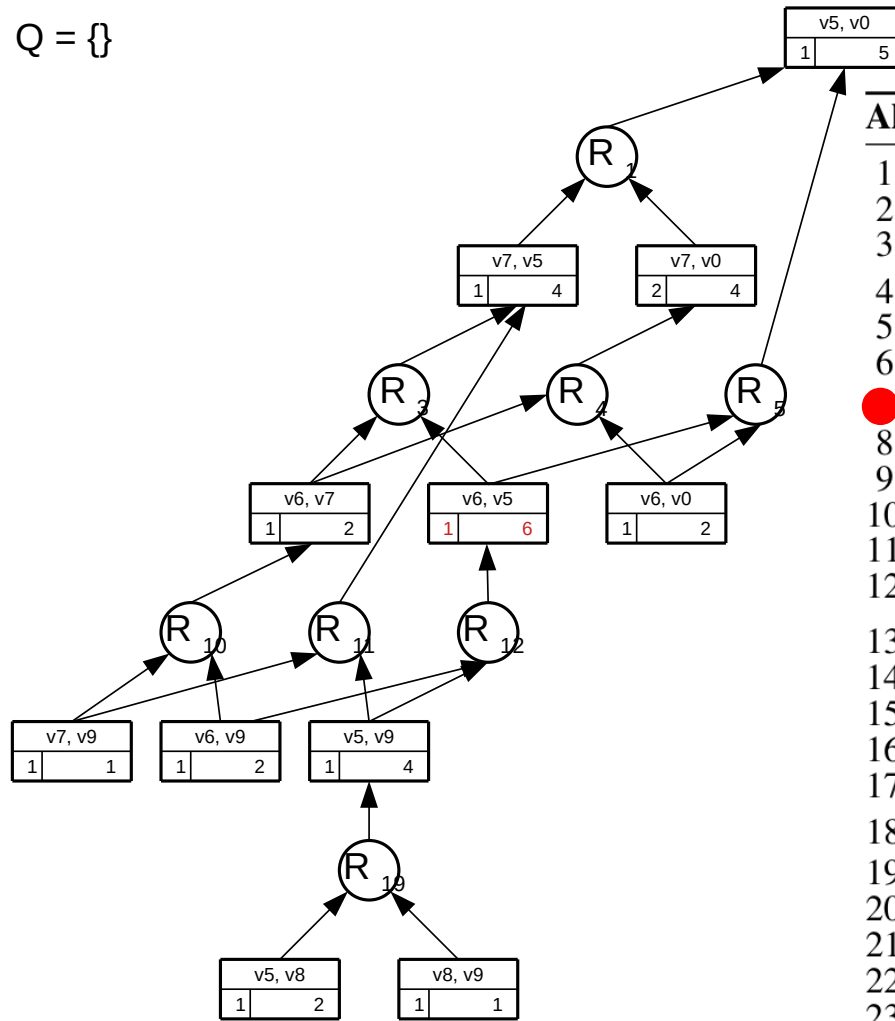
Algorithm 4 $DCH_{SCS}\text{-}WInc(G^*, G, e, k)$

```

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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  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:    $c_\phi(v_e) \leftarrow c_\phi(v_e) - 1$ ;
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  $v_r \in nbr^+(v_s)$  do
10:     $v'_s \leftarrow nbr^+(v_r)$ ;
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  $v_r \in nbr^+(v_s)$  do
15:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;  $v_{s3} \leftarrow nbr^+(v_r)$ ;
16:    if  $\phi(v_{s3}, G^*) = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
17:       $c_\phi(v_{s3}) \leftarrow c_\phi(v_{s3}) - 1$ ;
18:    $\phi \leftarrow \phi(s, G)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
19:   for each  $v_r \in nbr^-(v_s)$  do
20:     $v_{s1}, v_{s2} \leftarrow nbr^-(v_r)$ ;
21:    if  $\phi > \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
22:       $\phi \leftarrow \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$ ;  $c_\phi(v_s) \leftarrow 1$ ;
23:    else if  $\phi = \phi(v_{s1}, G^*) + \phi(v_{s2}, G^*)$  then
24:       $c_\phi(v_s) \leftarrow c_\phi(v_s) + 1$ ;
25:    $\phi(v_s, G^*) \leftarrow \phi$ ;

```

$\phi = 6$