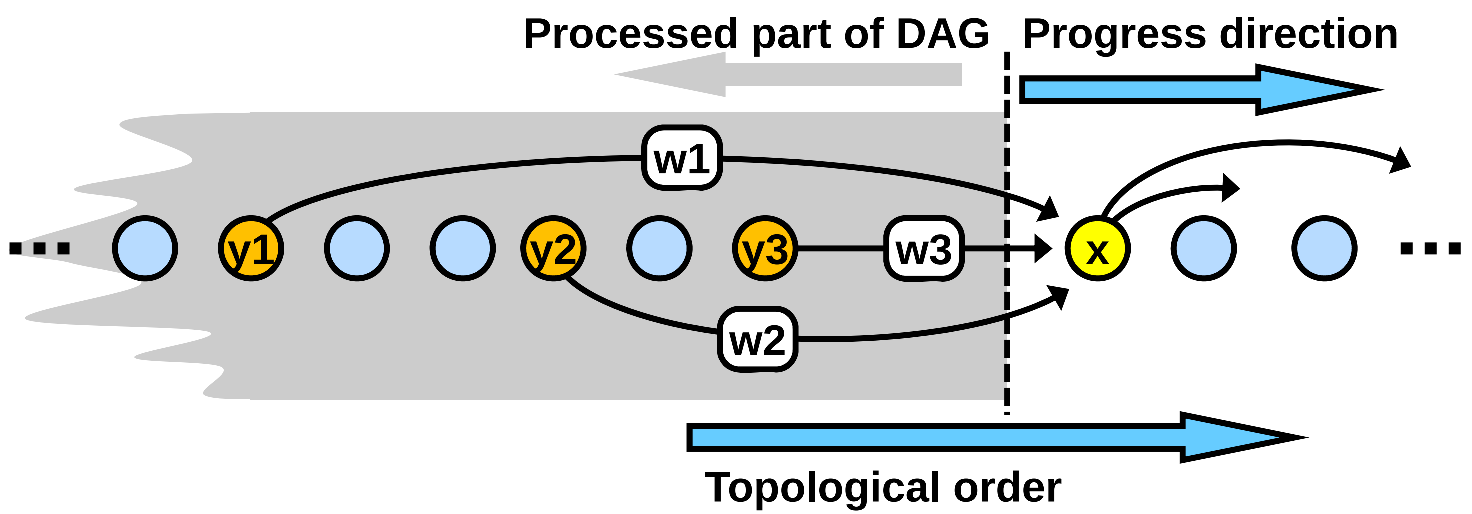
Longest path in DAG

We process nodes of DAG in their topological order.
Denote by d[x] length of the path which ends in x and its length is maximal.

Charakteristic DP view "from the end to the beginning":
-- d[x] is set when values of d are known for all previous (= already processed) nodes in the topological order.
-- d[x] is the maximum of values
{d[y1] + w1, d[y2] + w2, ..., d[yk] + wk},
where (y1, x), (y2, x) , ... are all edges ending in x
and w1, w2, ..., are their respective weights.

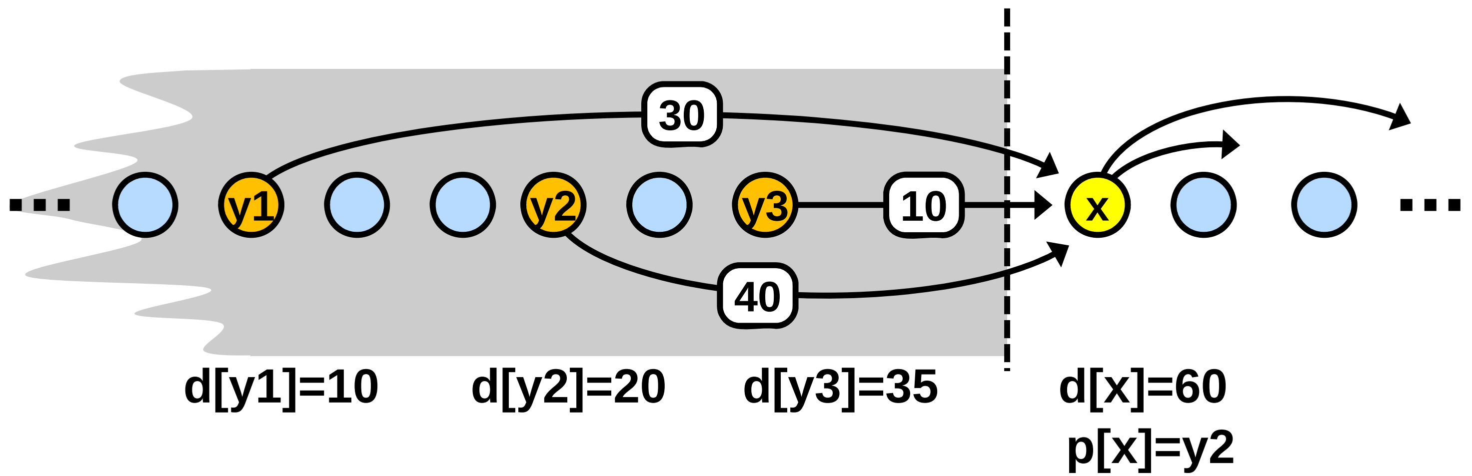


Longest path in DAG

-- d[x]is the maximum of values
{d[y1] + w1, d[y2] + w2, ..., d[yk] + wk},
where (y1, x), (y2, x) , ... are all edges ending in x
and w1, w2, ..., are their respective weights.

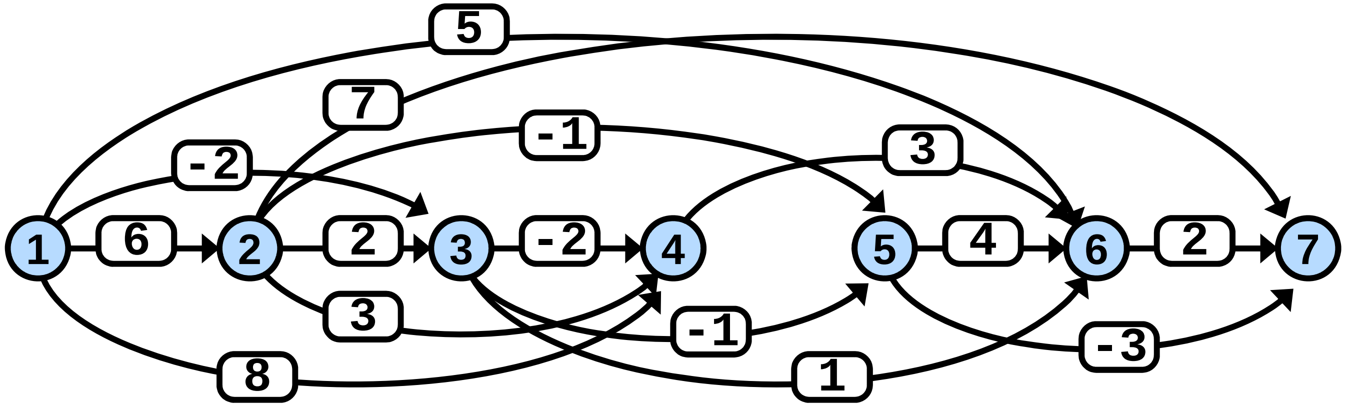
-- If all values {d[y1] + w1, d[y2] + w2, ..., d[yk] + wk}
are negative then none of them contributes to the longest path
and the value of d[x] is reset: d[x] = 0.

-- The node yj, for which the value d[yj] + wj is maximal and non-negative,
is set as a predecessor of x on the longest path.



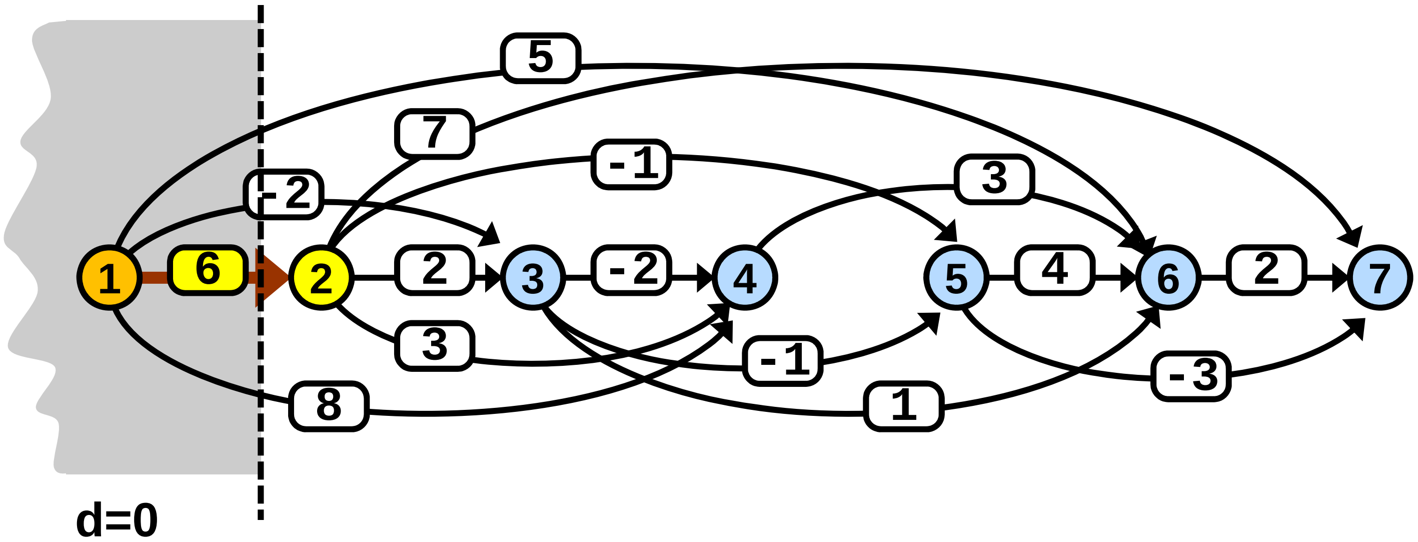
Longest path in DAG

Example



Find the longest path and its length.

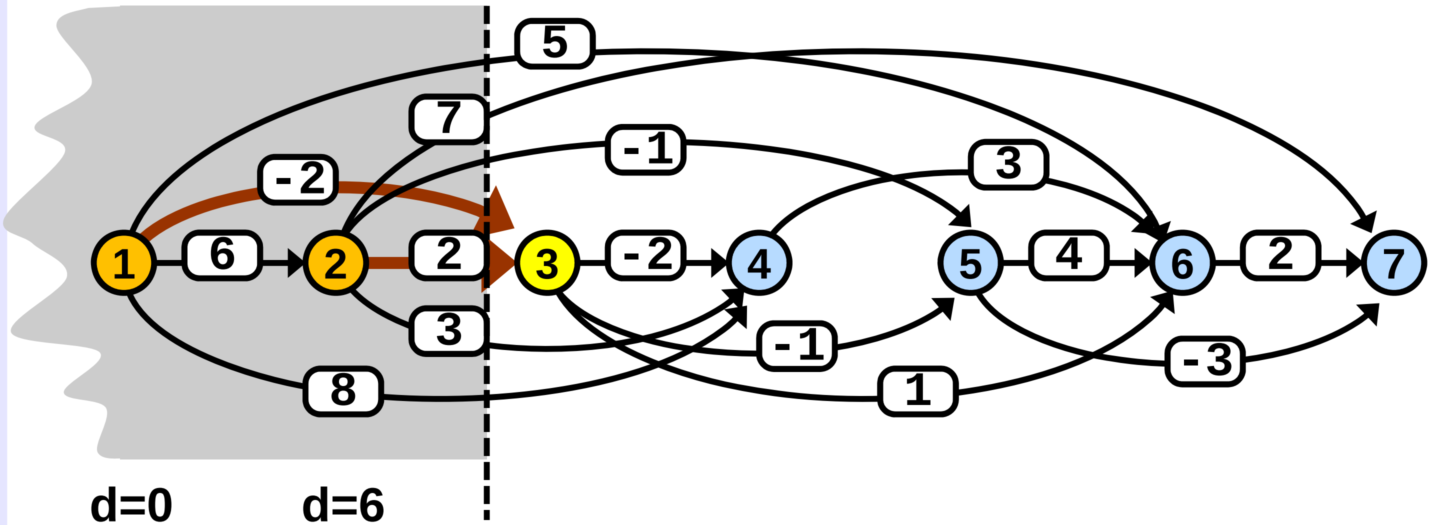
Longest path in DAG



d = max {0+6}
= 6

p=1

Longest path in DAG



d=0

p=nil

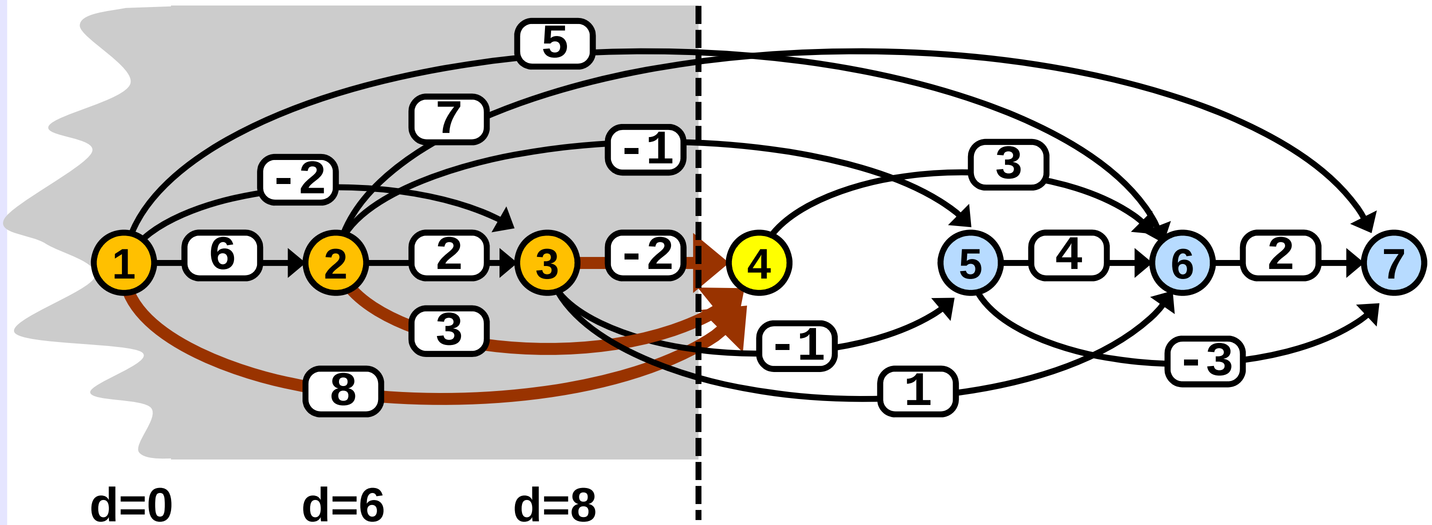
d=6

p=1

d = max {0+-2,
6+2}
= 8

p = 2

Longest path in DAG

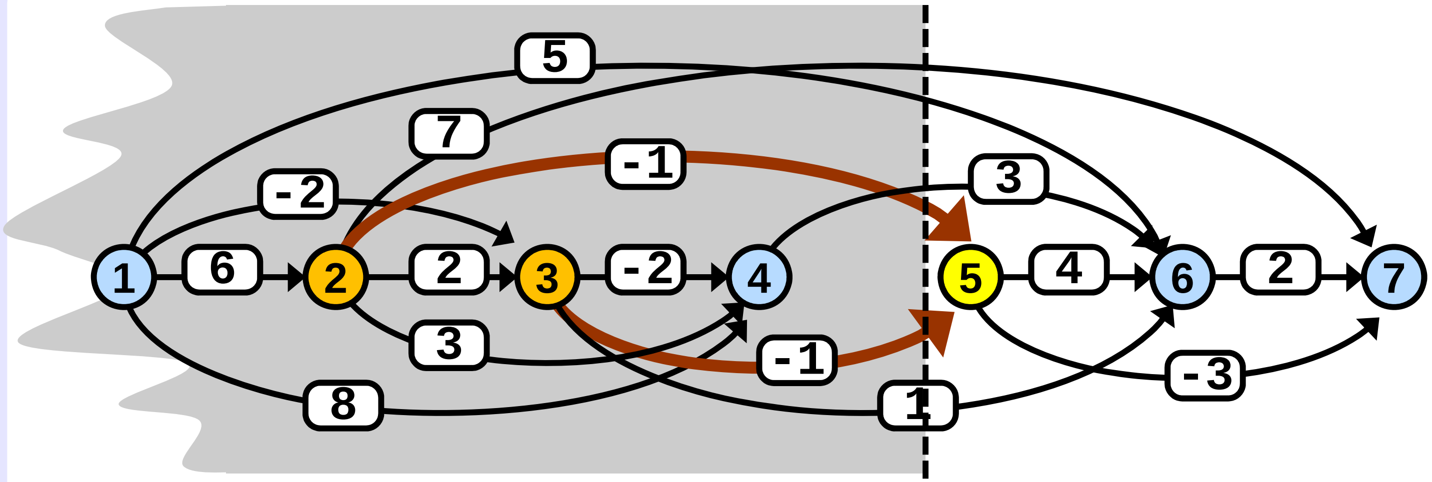


d=0	d=6	d=8
p=nil	p=1	p=2

$$d = \max\{0+8,\ 6+3,\ 8+-2\} = 9$$

$$p = 2$$

Longest path in DAG

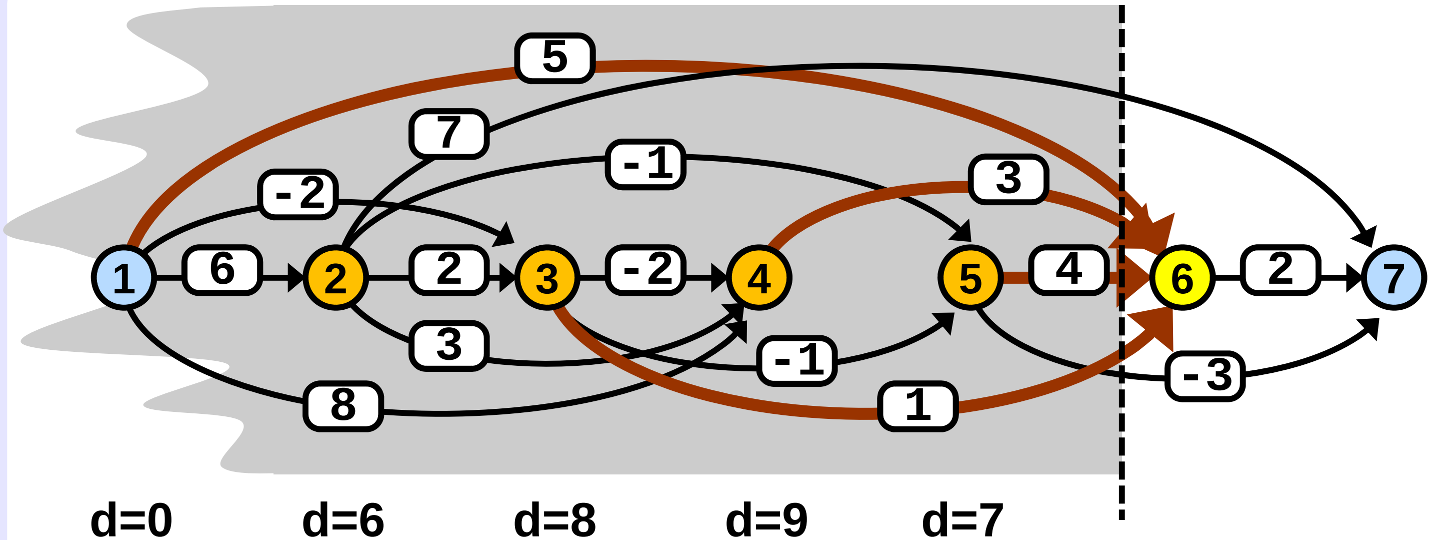


d=0	d=6	d=8	d=9
p=nil	p=1	p=2	p=2

d = max {6+-1,
8+-1}
= 7

p = 3

Longest path in DAG

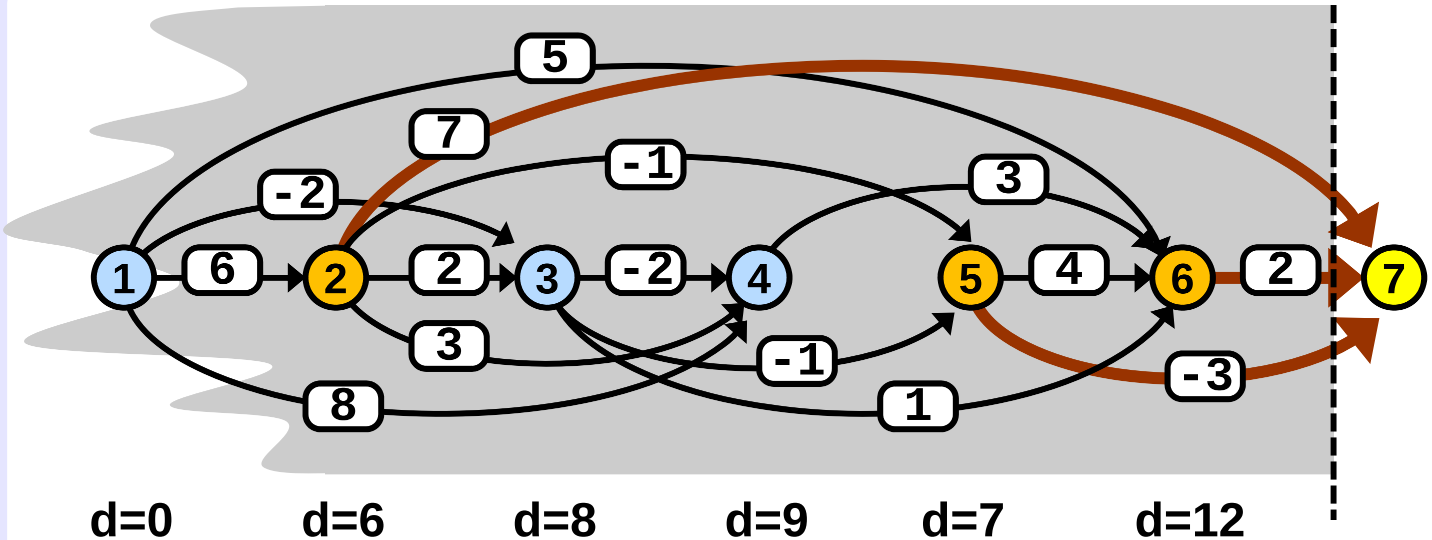


d=0	d=6	d=8	d=9	d=7
p=nil	p=1	p=2	p=2	p=3

d = max {0+5,
8+1,
9+3,
7+4}
= 12

p = 4

Longest path in DAG

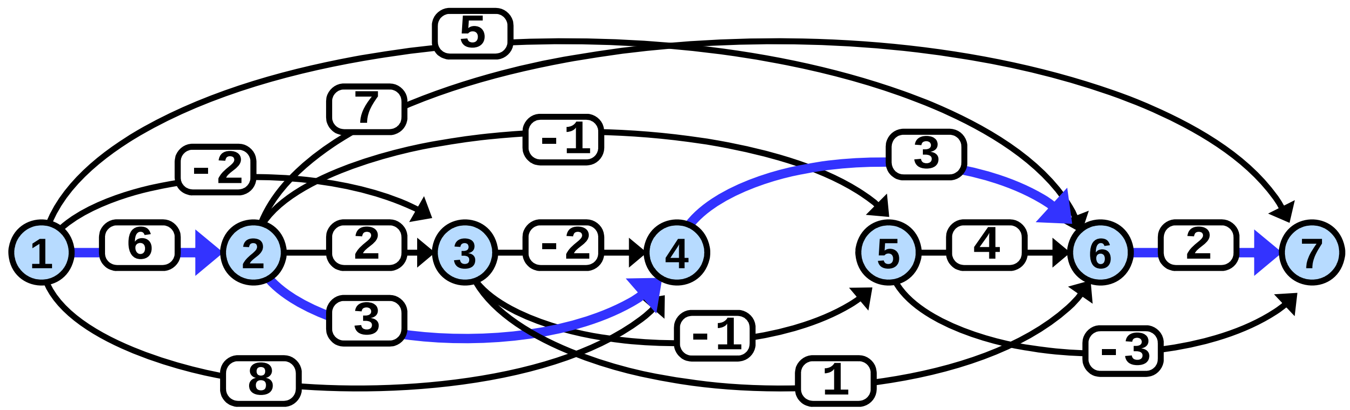


d=0	d=6	d=8	d=9	d=7	d=12
p=nil	p=1	p=2	p=2	p=3	p=4

d = max {6+7,
7+-3,
12+2}
= 14

p = 6

Longest path in DAG



d=0	d=6	d=8	d=9	d=7	d=12	d=14
p=nil	p=1	p=2	p=2	p=3	p=4	p=6

Length of the longest path: 14
The longest path itself: 1 -- 2 -- 4 -- 6 -- 7

Longest path in DAG

```
0. allocate memory for distance and predecessor of each node

1. for each x in V(G) {
       x.dist = negInfinity
       x.pred = null
   }

// supposing nodes are processed
// in ascending topological order
2. for each node x in V(G)
      for each edge e = (y, x) in E(G)
         if (x. dist < y.dist + e.weight) {
            x. dist = y.dist + e.weight
            x.pred = y;
   }
   if (x. dist < 0) x.dist = 0;     // avoid negative path lengths
}
```

0. Complexity Θ(N)

1. Complexity Θ(N)

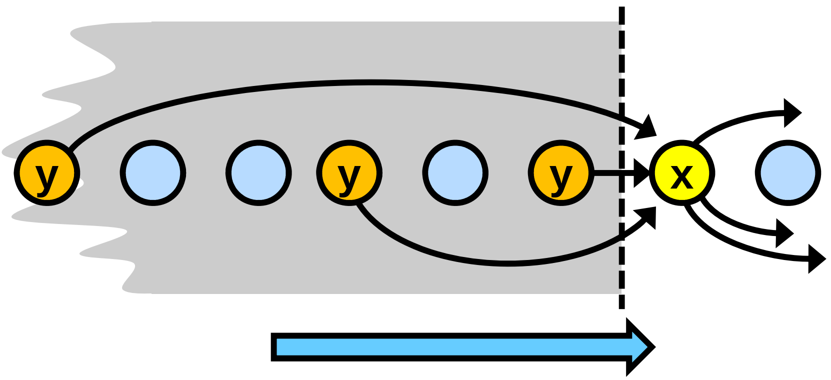
2. Complexity Θ(M), each edge is visited exactly once and it is processed in constant time.

Complexity: Θ(N+M)

Longest path in DAG

Variant I

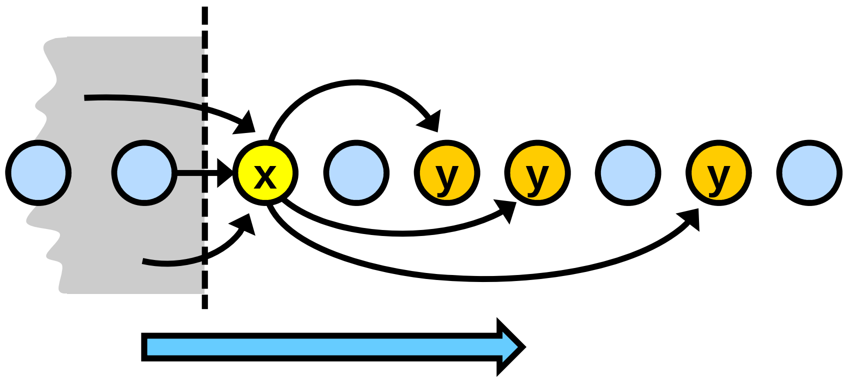
```
2. for each node x in V(G) {
      for each edge e = (y, x) in E(G)
         if (x. dist < y.dist + e.weight) {
            x. dist = y.dist + e.weight
            x.pred = y;
    }
    if (x. dist < 0) x.dist = 0;
}
```



order of processing
= topological order

Variant II

```
2. for each node x in V(G) {
      if (x. dist < 0) x.dist = 0;
      for each edge e = (x, y) in E(G)
         if (y. dist < x.dist + e.weight) {
            y. dist = x.dist + e.weight
            y.pred = x;
    }
}
```



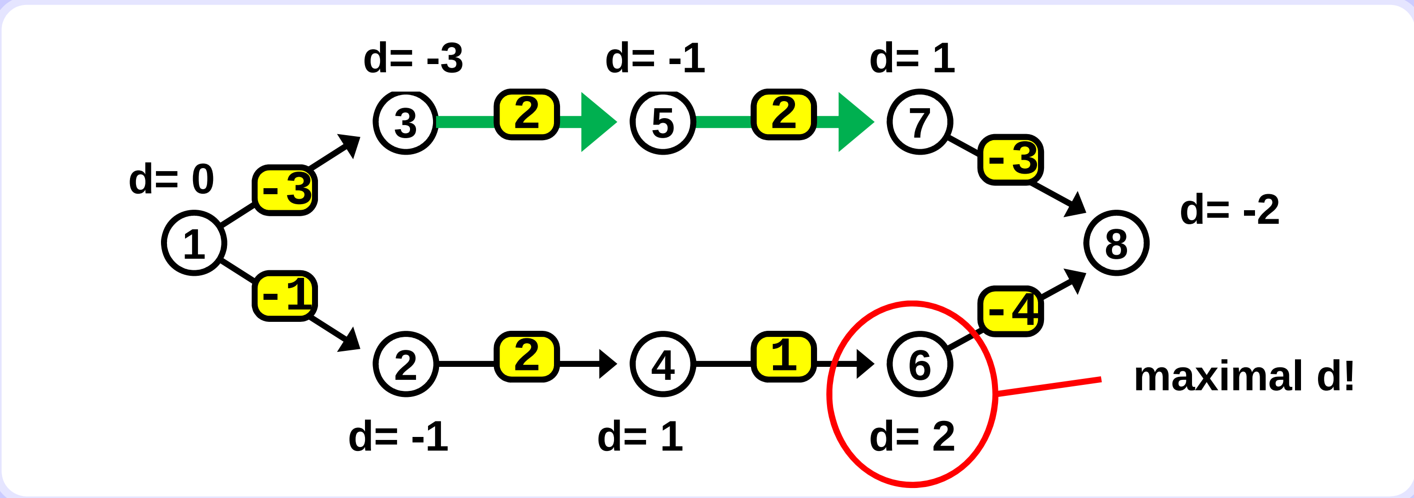
order of processing
= topological order

Longest path in DAG

Warning

Algorithms presented in the literature and on the web often solve the maximum path in DAG problem only for non-negative edge weights and do not mention explicitely this limitation.
Those algorithms cannot handle DAG containing negative weight edges.

Incorrect result produced by algorithm expecting only non-engative edge weights

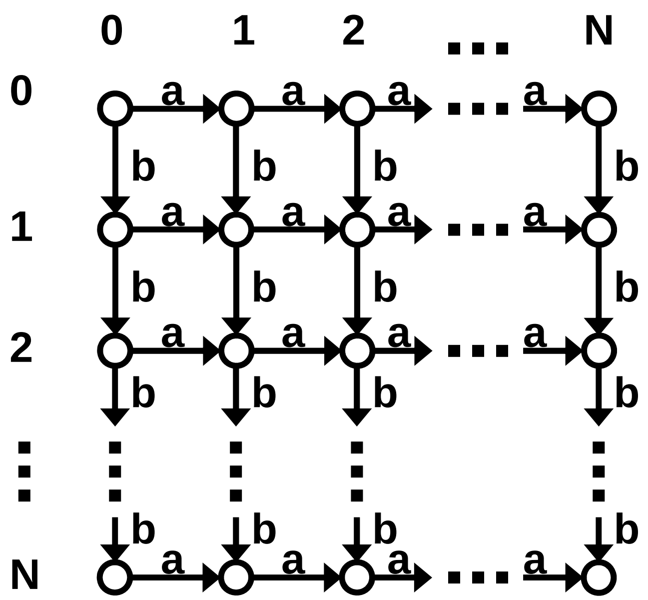


Actual maximum path is 3 -- 5 -- 7 which weight is 4.
Algorithm limited to non-negative weights finds only suboptimal path 1 -- 2 -- 4 -- 6 which weight is 2.

Longest path in DAG

Problem of reconstucting optimal paths -- the number of paths can be too high.

Example



Each path from the root to the leaf is optimal, its weight is N·(a+b).

Number of all paths is Comb(2N, N), it holds that 2^N < Comb(2N, N) < 4^N.

The numbert of optimal paths thus grows exponentially with the value of N.

N	# of optimal paths
1	2
10	184756
20	137846528820
30	118264581564861424
40	107507208733336176461620