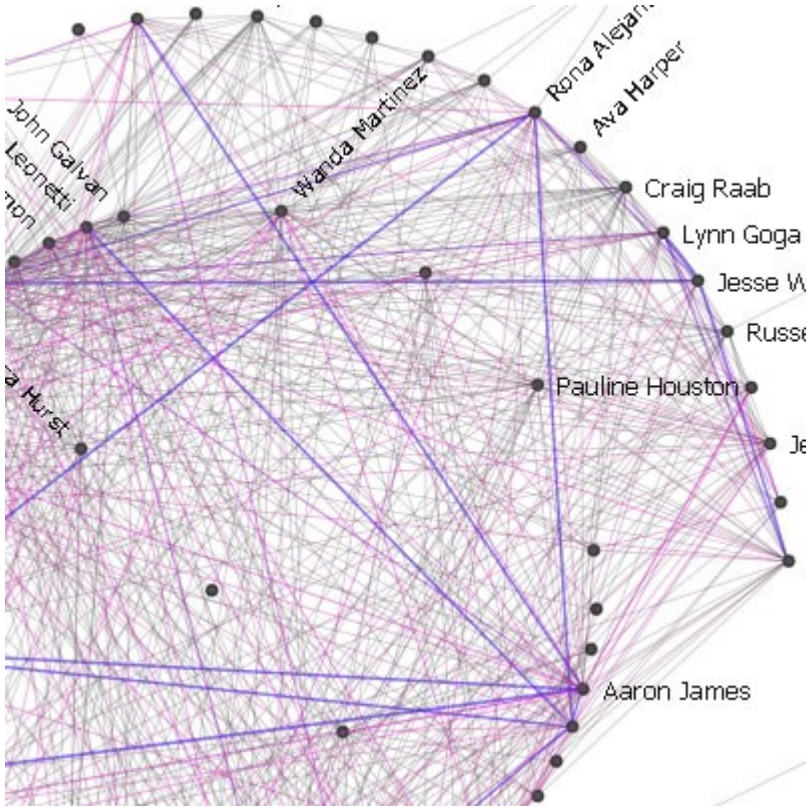
Programovacie techniky

10. Hľadanie do hĺbky, hľadanie do šírky, najkratšia cesta v grafe, Dijkstrov algoritmus

Facebook ako graf



Uzol = používateľ
Hrana = priateľstvo medzi používateľmi

Hľadanie do šírky?
Priemerný stupeň uzla?
Najkratšia cesta?
Priemer grafu?
- six degrees of separation

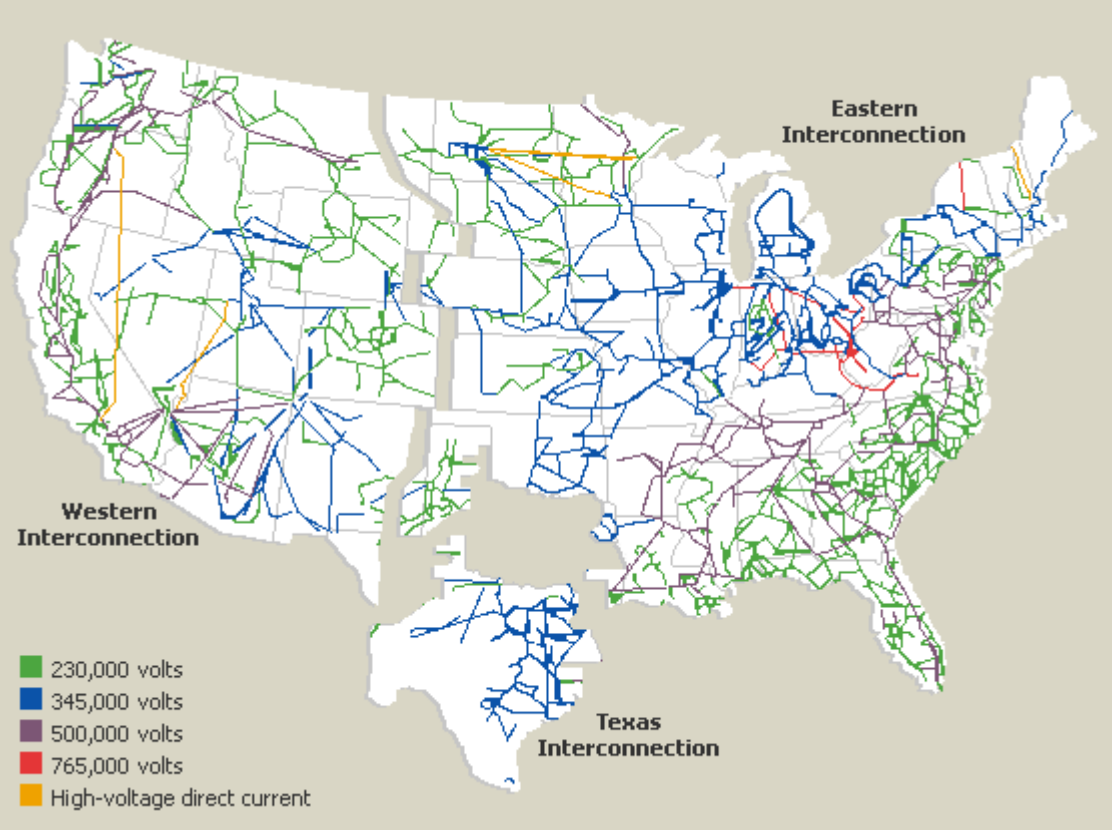
Six degrees of separation

"We found that six degrees actually overstates the number of links between typical pairs of users: while 99.6 percent of all pairs of users are connected by paths with 5 degrees, 92 percent are connected by only four degrees."

Six degrees of separation is the idea that all people on average are six, or fewer, social connections away from each other.

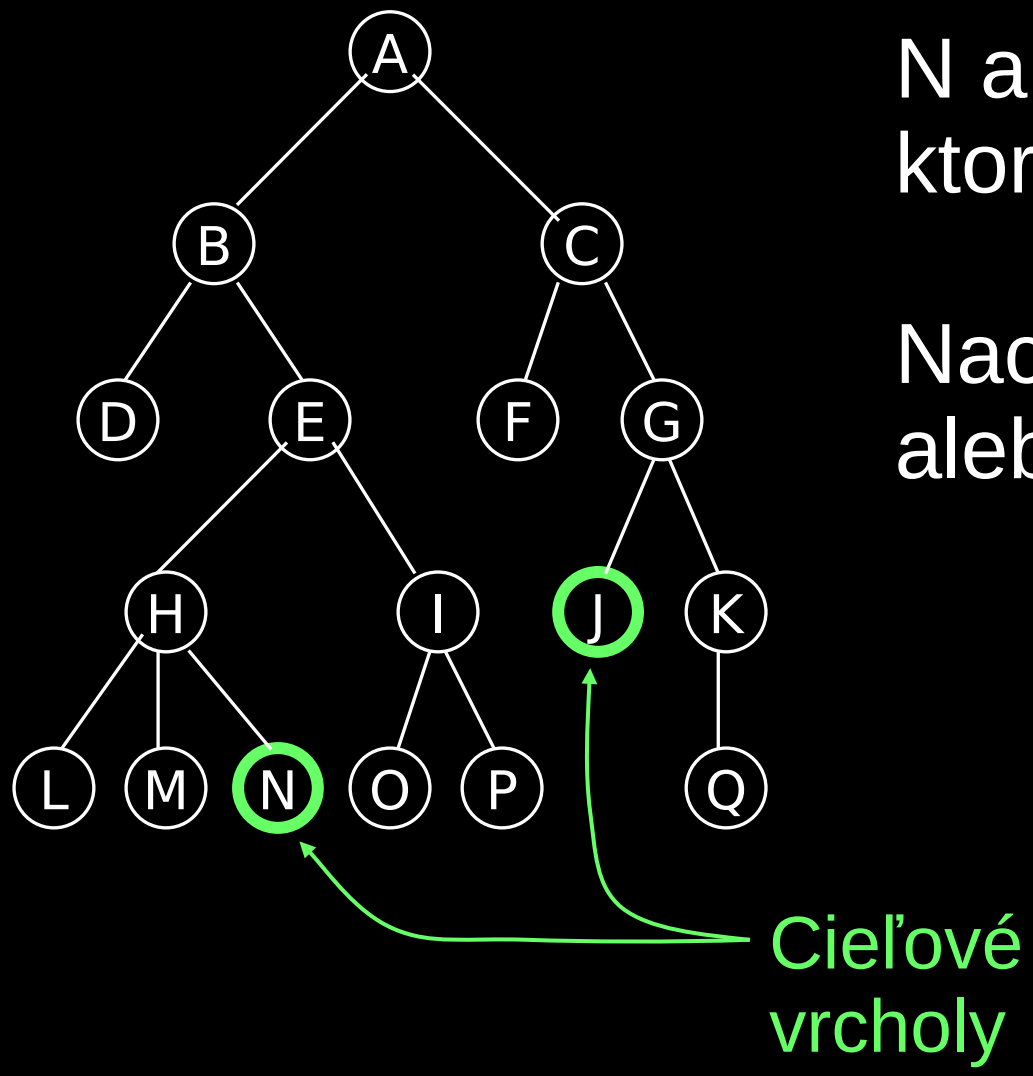
Source: https://www.facebook.com/notes/facebook-data-team/anatomy-of-facebook/10150388519243859

Elektrizačná prenosová sieť



Aký je priemerný stupeň uzla?
V porovnaní s Facebookom?

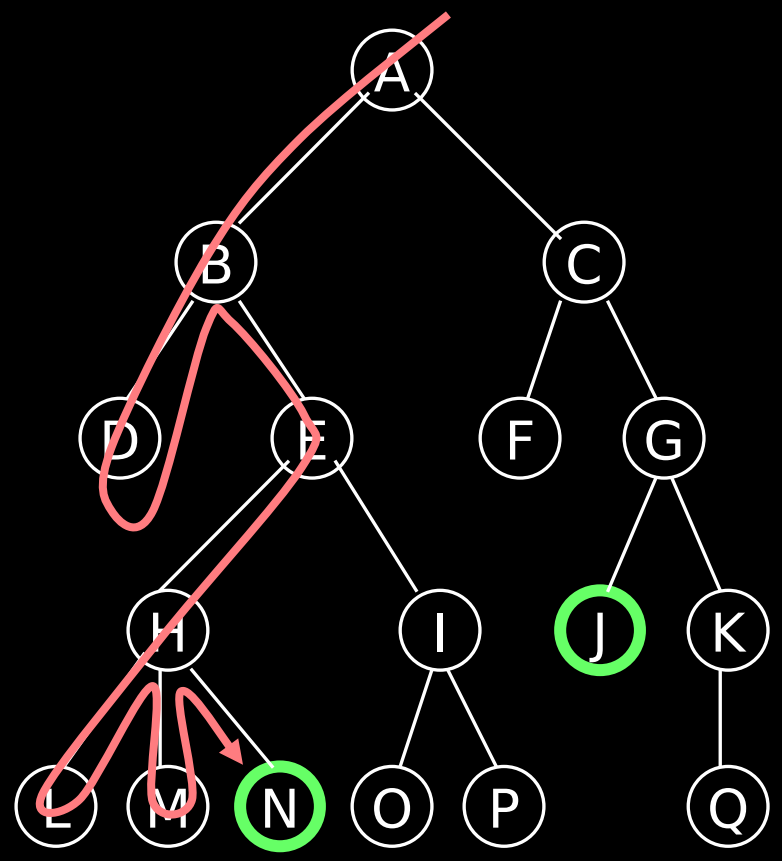
Hľadanie v strome



N a J sú vrcholy, ktoré hľadáme

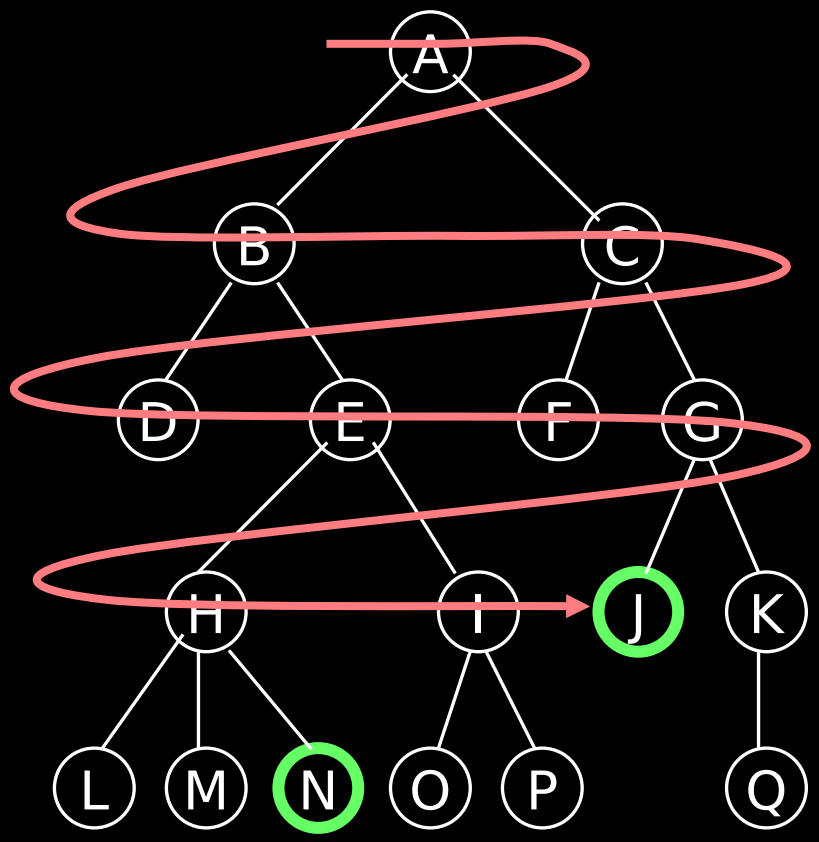
Nachádza sa N alebo J v strome?

Hľadanie do hĺbky



Vrcholy sú navštívené v poradí: A B D E H L M N I O P C F G J K Q

Hľadanie do šírky



Vrcholy sú navštívené v poradí: A B C D E F G H I J K L M N O P Q

J je nájdené pred N

Hľadanie do hĺbky

```
put the root node on a STACK;

while (stack is not empty) {
    remove a node from the stack;

    if (node is a goal node) return success;

    put all children of node onto the stack;
}

return failure;
```

Hľadanie do šírky

```
put the root node on a QUEUE;

while (queue is not empty) {
    remove a node from the queue;

    if (node is a goal node) return success;

    put all children of node onto the queue;
}

return failure;
```

Poznámky

Pre veľký strom, hľadanie do šírky môže mať veľké nároky na pamäť

Pre veľký strom, hľadanie do hĺbky môže trvať dlho pre cieľový uzol blízko koreňa

Zložitosť DFS, BFS: $O(|E|)$

$|E|$ je kardinalita (mohutnosť) množiny E, t. j. počet hrán

DFS, BFS

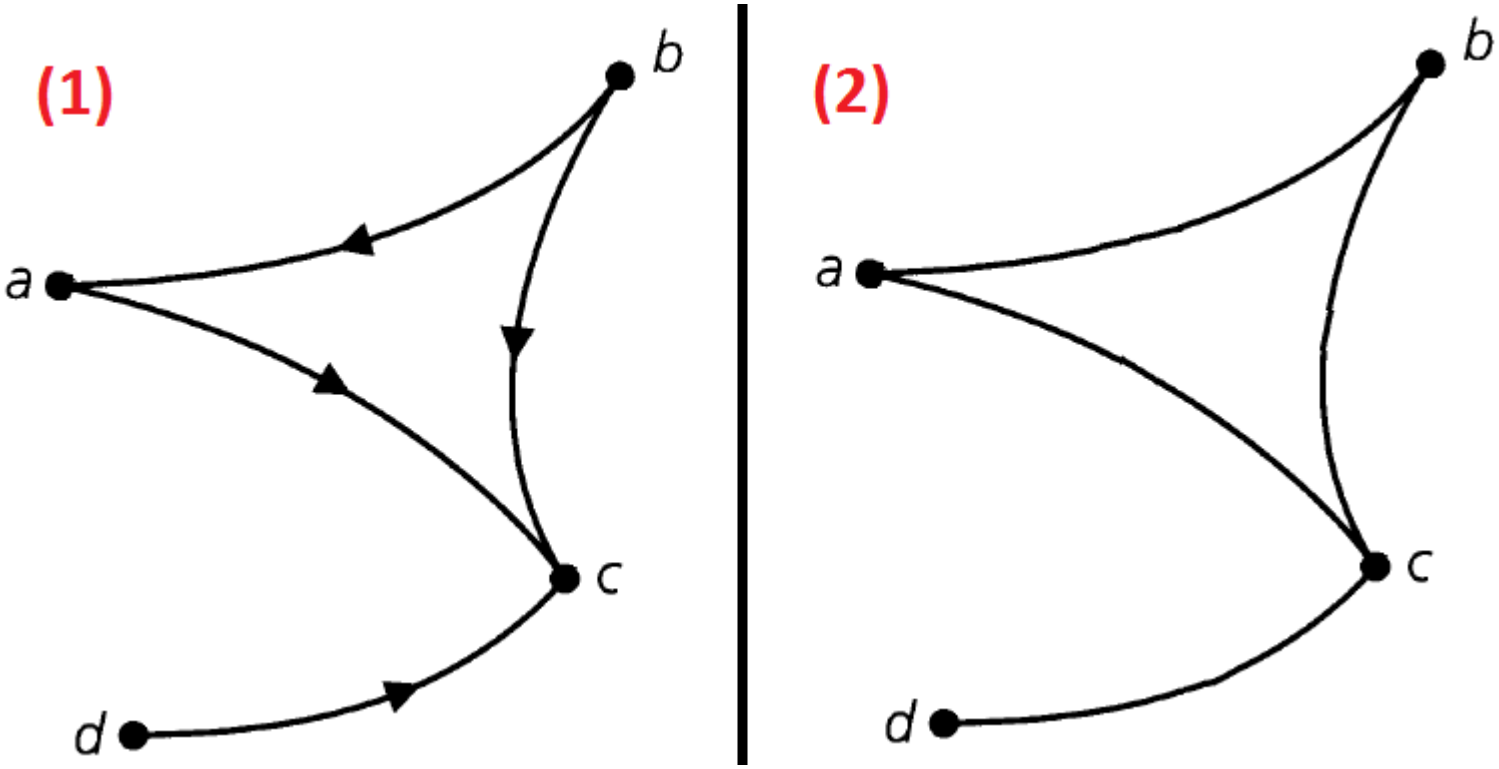
BFS = breadth first search

DFS = depth first search

Úloha

Implementuj DFS, ktorý hľadá len do hĺbky depth_limit

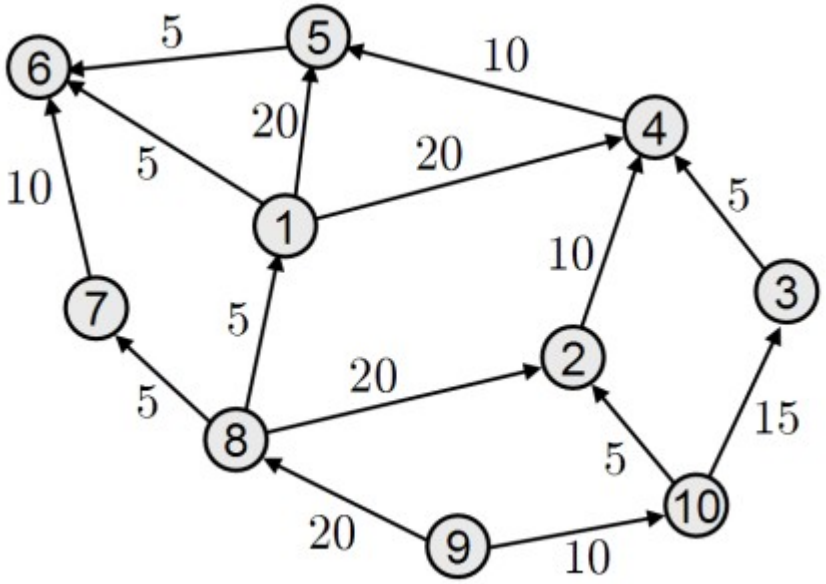
Orientovaný graf



Directed graph
Orientovaný graf

Undirected graph
Neorientovaný graf

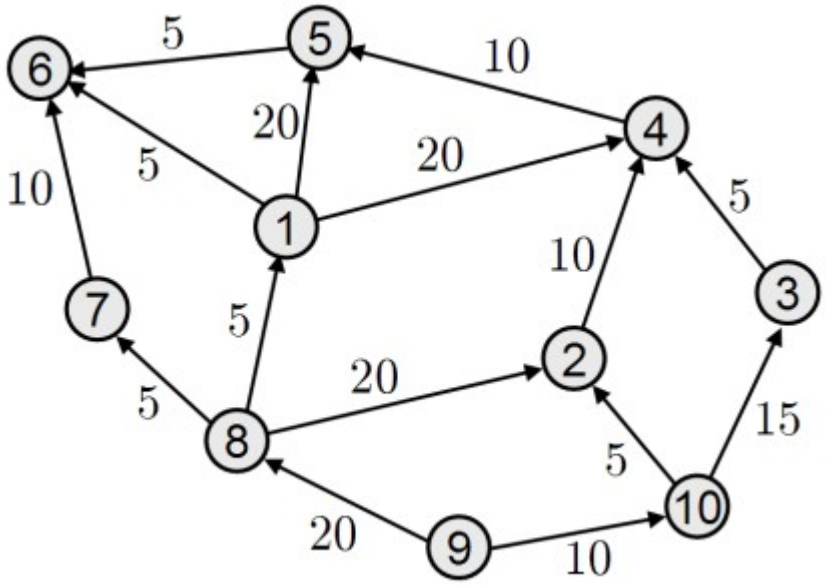
Hranovo-ohodnotený graf



Orientovaný hranovo-ohodnotený graf

Aká je najkratšia cesta z vrcholu 10 do vrcholu 6?

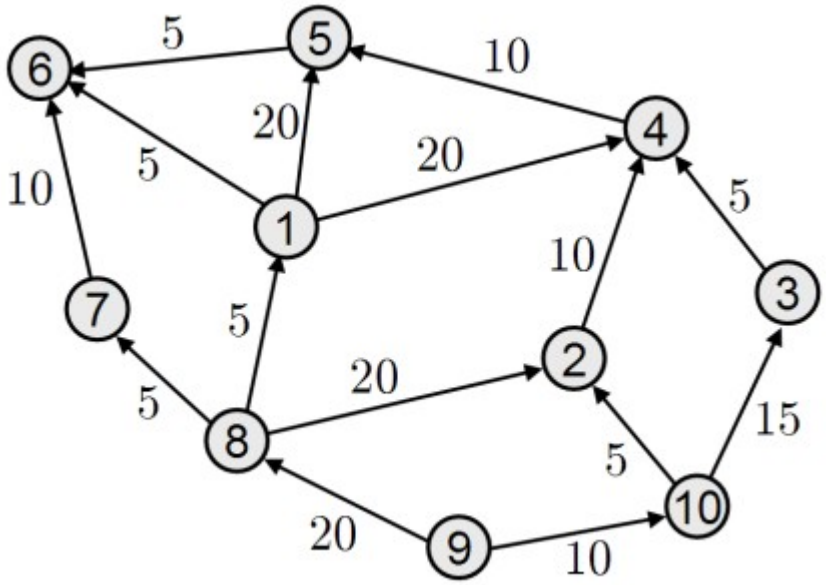
Hranovo-ohodnotený graf



G = (V, E, w) alebo G = (V, E)

w je funkcia E—>R, kde R je množina reálnych čísel
V je množina vrcholov
E je množina hrán: V x V

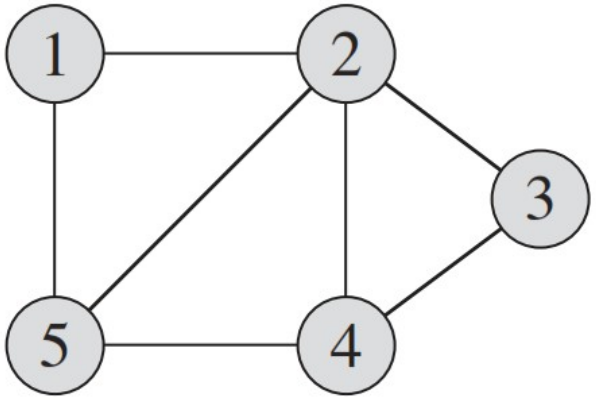
Hranovo-ohodnotený graf

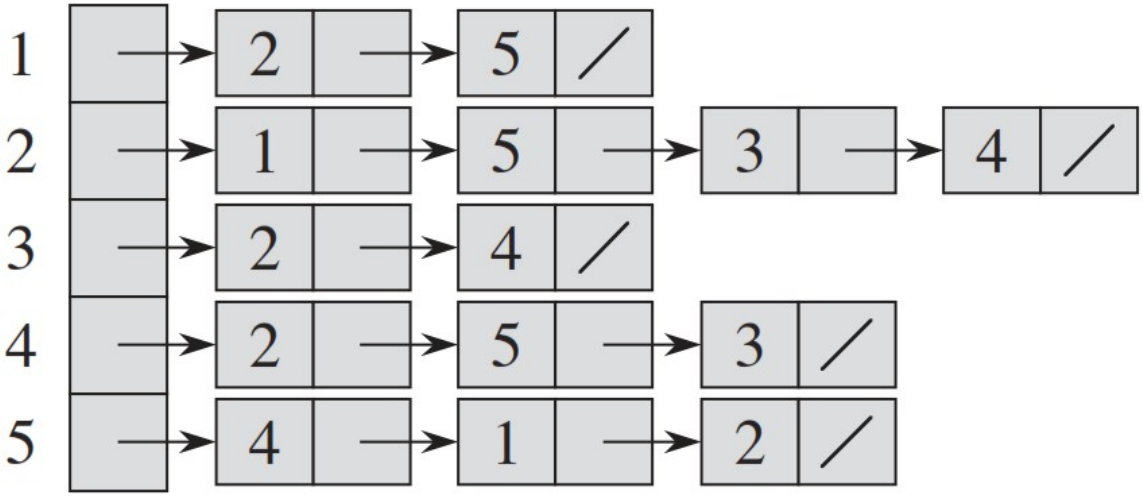


G = (V, E, w) alebo G = (V, E)

V = vertices
E = edges

Reprezentácia grafu



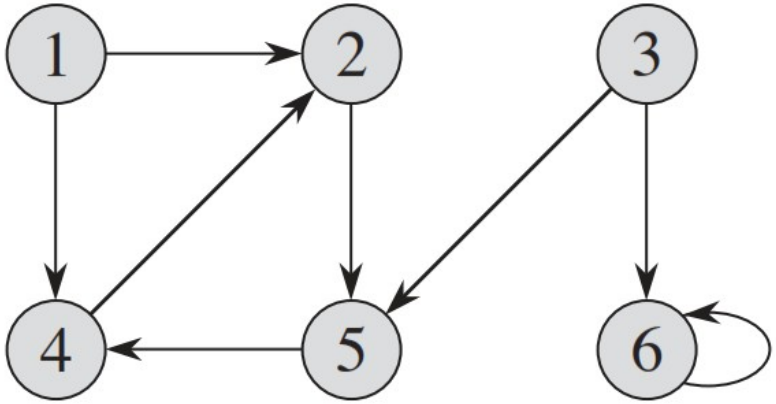


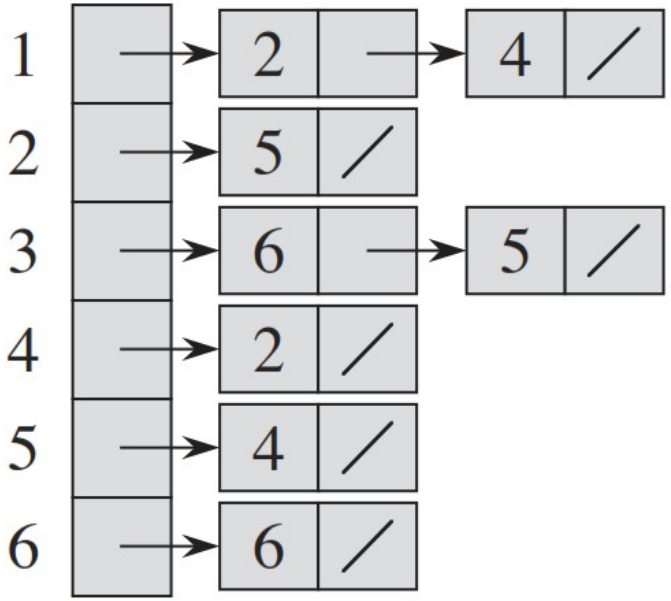
Zoznam susedov
Adjacency list

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

Matica susedov
Adjacency matrix

Orientovaný graf





Zoznam susedov
Adjacency list

	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

Matica susedov
Adjacency matrix

BFS: pseudocode

```
BFS(G, s)
for each vertex u ∈ G.V − {s}
    u.color = WHITE
    u.d = ∞
    u.π = NIL
s.color = GRAY
s.d = 0
s.π = NIL
Q = ∅
ENQUEUE(Q, s)
while Q ≠ ∅
    u = DEQUEUE(Q)
    for each v ∈ G.Adj[u]
        if v.color == WHITE
            v.color = GRAY
            v.d = u.d + 1
            v.π = u
            ENQUEUE(Q, v)
    u.color = BLACK
```

Farba = {white, gray, black}

white nenájdený

gray nájdený

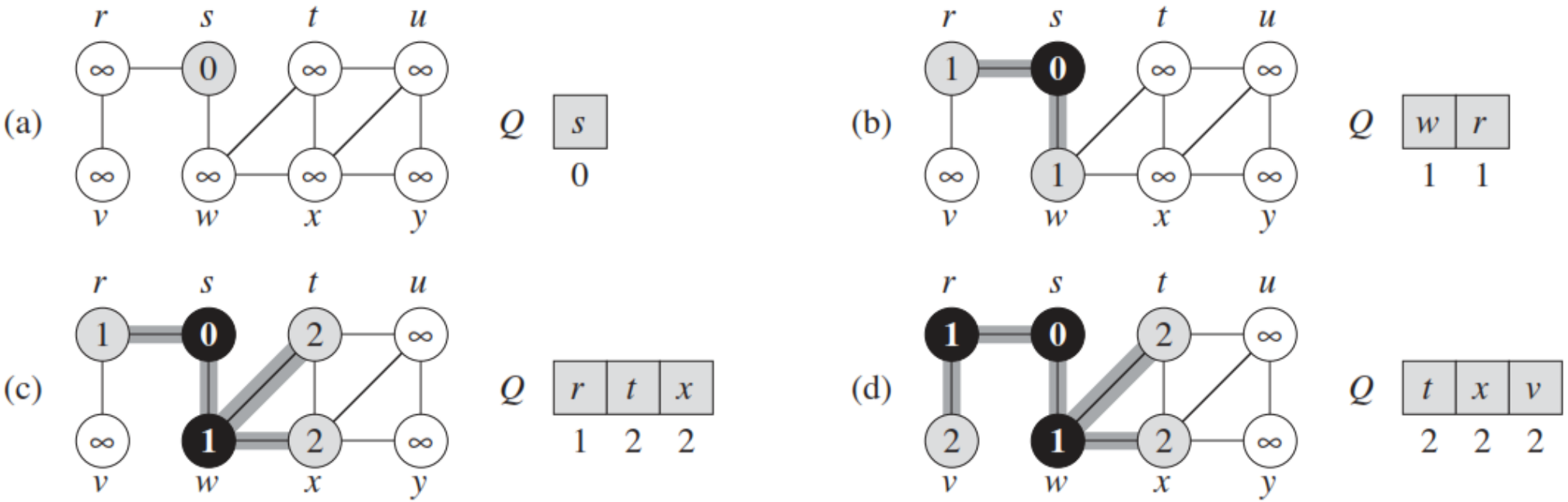
black všetci susedia nájdení

u.d je vzdialenosť z s do u

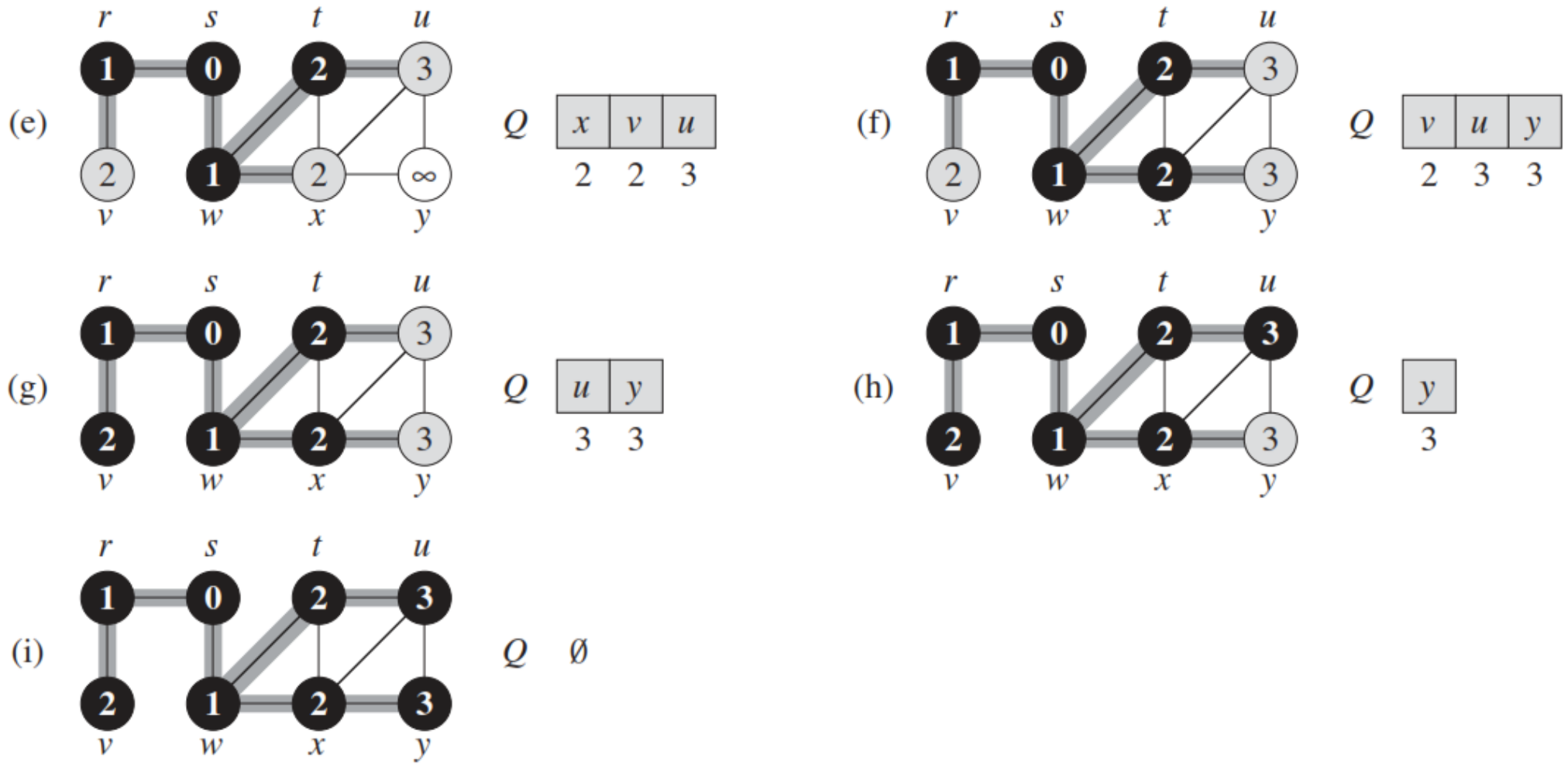
u.π je predchodca u (predecessor)

Q je fronta (queue)

BFS: príklad

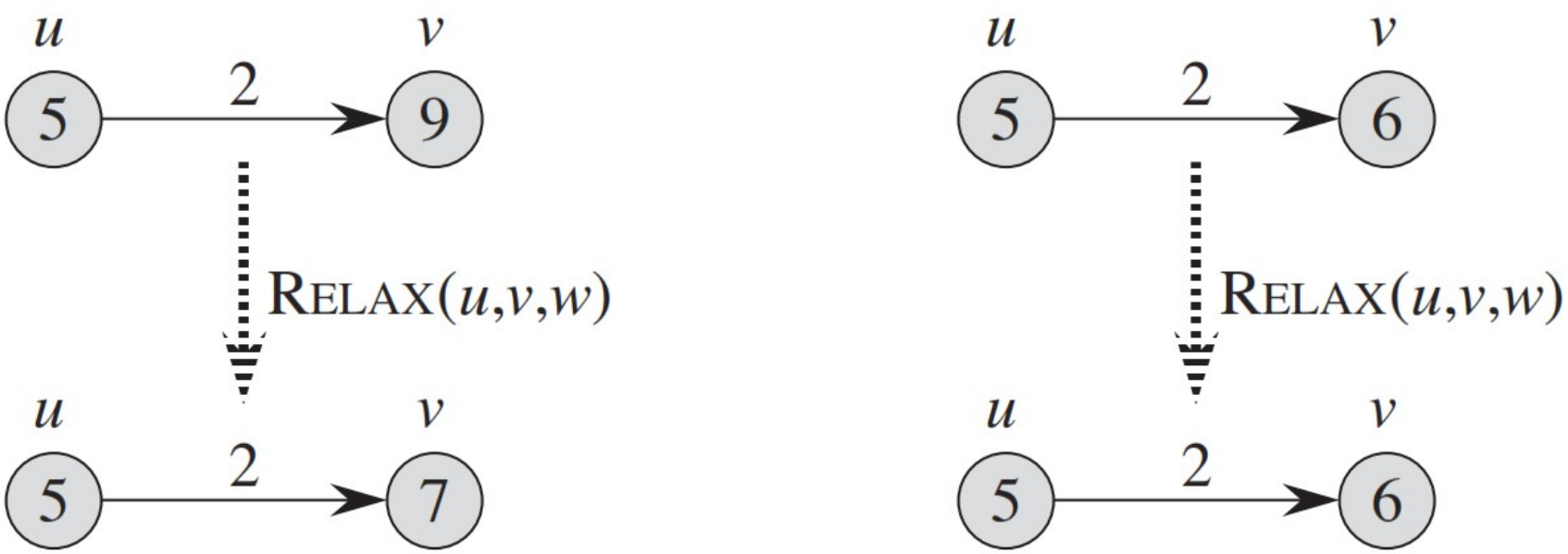

(a)
r s t u
∞ 0 ∞ ∞
∞ ∞ ∞ ∞
v w x y
Q s
0
(b)
r s t u
1 0 ∞ ∞
∞ 1 ∞ ∞
v w x y
Q w r
1 1
(c)
r s t u
1 0 2 ∞
∞ 1 2 ∞
v w x y
Q r t x
1 2 2
(d)
r s t u
1 0 2 ∞
2 1 2 ∞
v w x y
Q t x v
2 2 2

BFS: príklad


(e)
r s t u
1 0 2 3
2 1 2 ∞
v w x y
Q x v u
2 2 3
(f)
r s t u
1 0 2 3
2 1 2 3
v w x y
Q v u y
2 3 3
(g)
r s t u
1 0 2 3
2 1 2 3
v w x y
Q u y
3 3
(h)
r s t u
1 0 2 3
2 1 2 3
v w x y
Q y
3
(i)
r s t u
1 0 2 3
2 1 2 3
v w x y
Q Ø

Najkratšia cesta: relaxácia

```
RELAX(u, v, w)
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```



Najkratšia cesta: inicializácia

```
INITIALIZE-SINGLE-SOURCE(G, s)
for each vertex v ∈ G.V
    v.d = ∞
    v.π = NIL
s.d = 0
```

Dijkstrov algoritmus

```
DIJKSTRA(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

S je množina vrcholov, ktorých vzdialenosť od s je už známa

Q je min priority queue podľa vzdialenosti uzla od s

w je nezáporné ohodnotenie hrany

Zložitosť

```
DIJKSTRA(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

G je orientovaný graf

O(|V|)

minimum: O(log |V|)

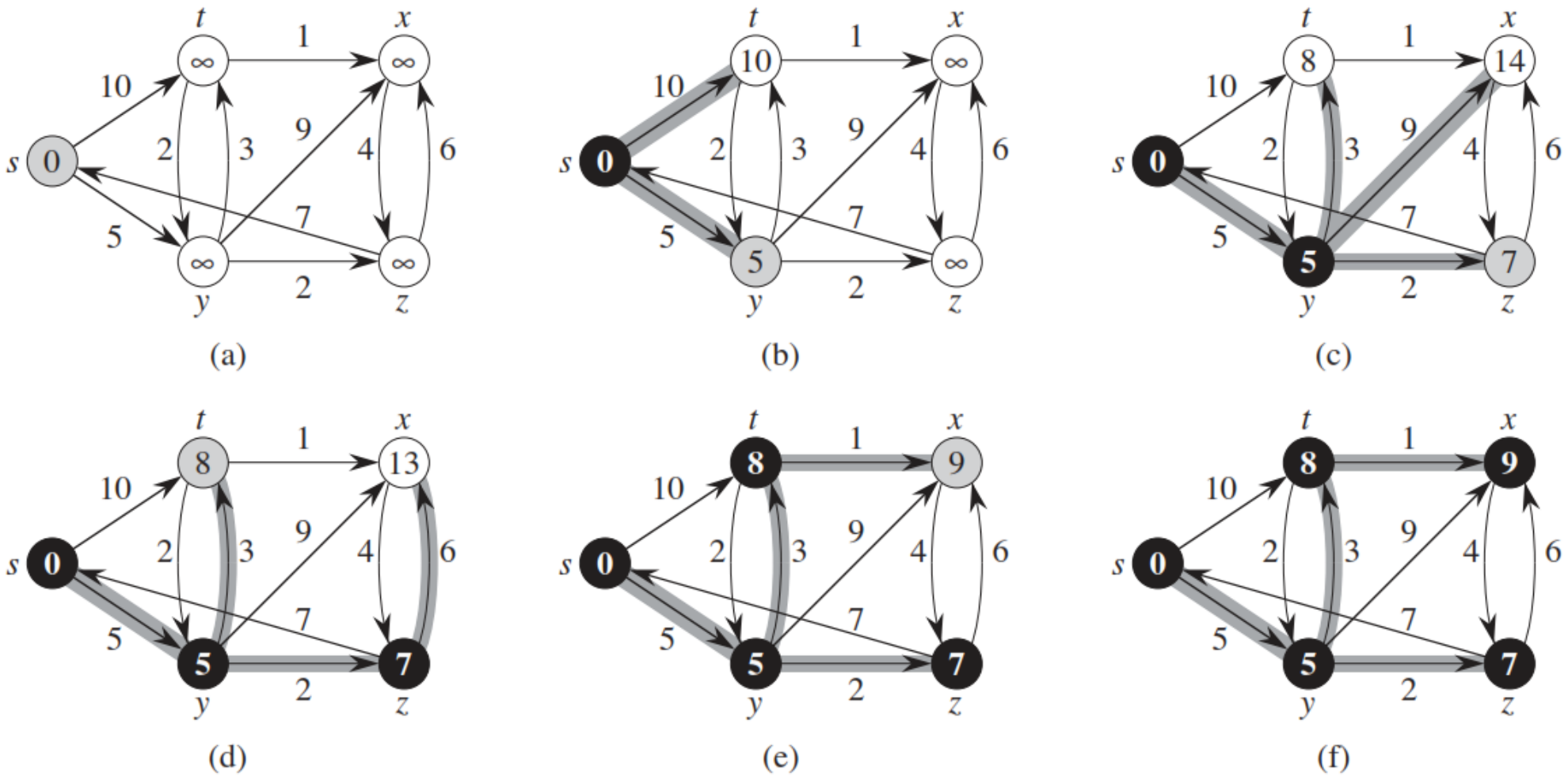
O(|E|), celkovo

decrease key: zmena hodnoty vzdialenosti, O(log |V|)

Heap: O((|E|+|V|) log |V|)

Fibonacci heap: O(|E| + |V| log |V|)

Dijkstrov algoritmus



Edsger Wybe Dijkstra

Najkratšia cesta: 1959, Holandsko

Dynamické programovanie

Dynamické programovanie:

- Dijkstrov algoritmus
- Výpočet Fibonacciho čísla
- Zarovnanie reťazcov
- A mnohé ďalšie

Nie je dynamické programovanie:

- Quicksort
- Merge sort

Dynamické programovanie

Dynamické programovanie = riešenie problému pomocou rekurzívneho delenia na podproblémy, a zároveň zapamätanie úspešnosti riešenia týchto podproblémov. Zapamätanie úspešnosti neviem v prípade Quicksort a Merge sort využiť, pretože riešenia nemajú žiadny prekryv.

Richard Bellman

Dynamické programovanie: 1949-55, USA.

Úlohy

Príklad, keď Dijkstra nenájde najkratšiu cestu v grafe so zápornými hodnotami hrán?

Aký je výstupný stupeň každého vrcholu? Výstupný stupeň v orientovanom grafe je počet hrán vychádzajúcich z vrcholu.

Vypočítaj priemer (diameter) grafu. Priemer je najdlhšia najkratšia cesta v grafe.

Timeline

1730: Stirling (UK): aproximácia faktoriálu
1945: von Neumann (USA): merge sort
1949-55: Belman (USA): dynamické programovanie
1955: „umelá inteligencia“ (John McCarthy, USA)
1957: Fortran (John Backus, USA)
1958: Lisp (John McCarthy, USA)
1959: Dijkstra (Holandsko): najkratšia cesta
1960: Hoare (UK): quick sort
1960: P.F. Windley, A.D. Booth, A.J.T. Colin, T.N. Hibbard (UK): binary search tree
1964: J. W. J. Williams (UK, Canada): heap sort
1972: Dennis Ritchie (USA): jazyk C
1983: Bjarne Stroustrup (Denmark, USA): jazyk C++
1995: James Gosling (USA): jazyk Java
1998: Larry Page, Sergey Brin (USA): Google