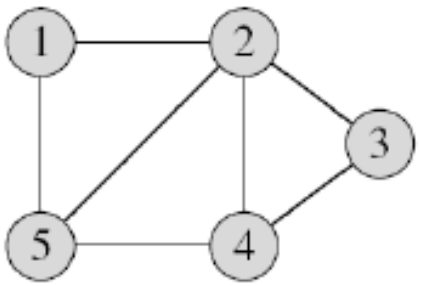
Grafovi

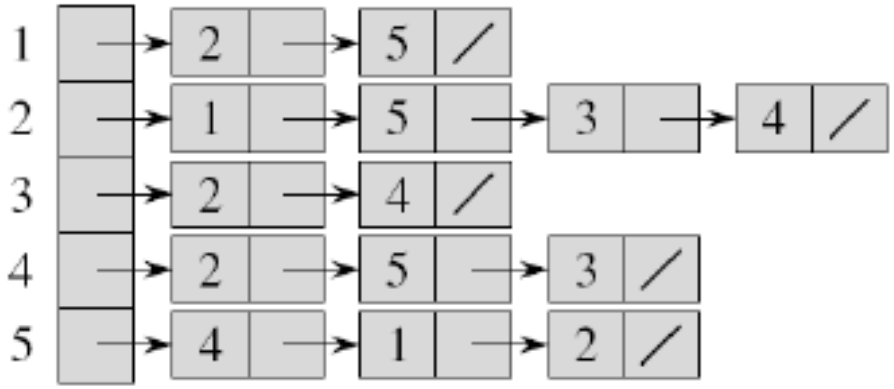
Strukture podataka i algoritmi 2

Grafovi

- $G = (V, E)$
- V – skup čvorova *(vertices)*
- E – skup veza *(edges)*
- 2 načina predstavljanja:
  - Kolekcija lista povezanosti
  - Matrica povezanosti
- Težinski grafovi



	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



Zadatak

- Web sajt treba organizovati tako da sve stranice budu lako dostupne sa početne.
- Dubina stranice – broj linkova koje treba pratiti da bi se od početne strane stiglo do tražene.
- Na ulazu se zadaje broj web strana (n) i broj njihovih povezanosti linkovima (l), veze su orijentisane.
- Zatim se zadaju povezanosti i maksimalna dozvoljena dubina.
- Odrediti strane čija je dubina veća od dozvoljene.

Pretraga u širinu

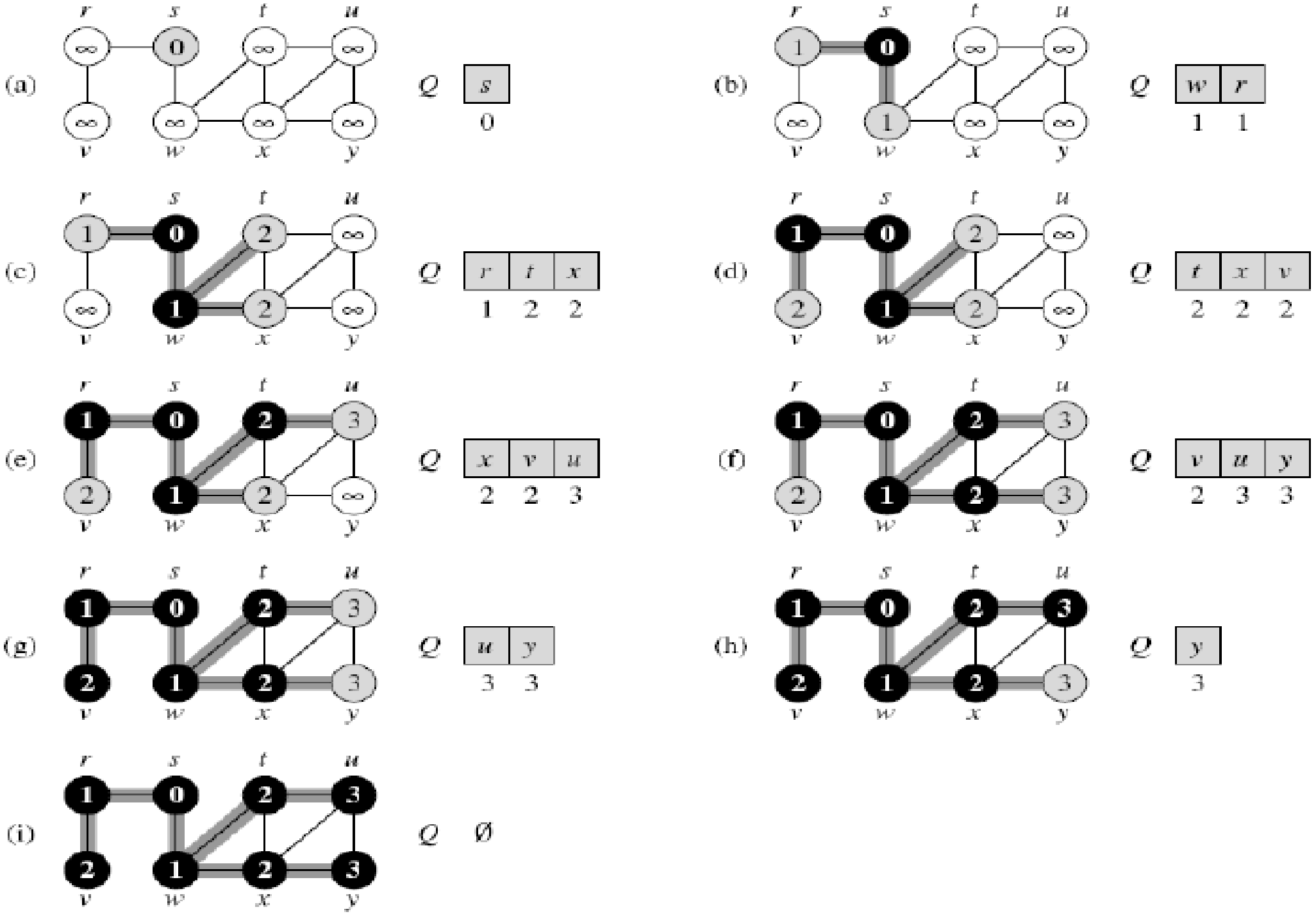
- Breadth first search – BFS
- Izvorni čvor s – 0. nivo
- Svi susedi čvora s – 1. nivo
- Za svaki čvor iz 1. nivoa, svi nerazmatrani susedi – 2. nivo itd.
- Širinsko stablo, koren s
- Najkraće rastojanje od čvora s do svakog čvora.

Pretraga u širinu

- $Color[u]$ – boje čvorova, bela, siva i crna
- $\pi[u]$ – roditelj čvora u
- $d[u]$ – rastojanje čvora u od početnog čvora
- FIFO lista Q

Pretraga u širinu

- $Color[u]$ – boje čvorova, bela, siva i crna
- $\pi[u]$ – roditelj čvora u
- $d[u]$ – rastojanje čvora u od početnog čvora
- FIFO lista Q
- Bela - $d[i] = \infty$
- Siva - $i \in Q$
- Crna - $d[i] \neq \infty,\ i \notin Q$


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

Pretraga u širinu

```
BFS( G , s )
   for each u ∈ V − {s}
      color[u] = WHITE
      d[u] = ∞
      π[u] = NULL
   color[s] = GRAY
   d[s] = 0
   π[s] = NULL
   Q = NULL
   ENQUEUE( Q , s )
```

```
while Q ≠ 0
   DEQUEUE( Q , u )
   for each v ∈ Adj[u]
      if color[v] = WHITE
         color[v] = GRAY
         d[v] = d[u] + 1
         π[v] = u
         ENQUEUE( Q , v )
   color[u] = BLACK
```

```
PRINT_PATH( G , s , v )
   if v = s
      print s
   else if π[v] = NULL
      print "Nema puta od s do v"
   else
      PRINT_PATH( G , s , π[v] )
      print v
```

Test primeri

```
14
16
0 1
0 2
0 4
1 3
1 5
2 7
2 8
4 6
4 2
3 2
5 4
6 9
6 10
2 11
10 12
10 13
0
3
```

```
13
19
0 1
0 2
0 3
0 6
1 8
2 4
2 9
3 4
4 9
4 10
4 7
4 5
5 7
6 7
7 10
7 11
8 11
9 10
10 12
0
2
```

Test primeri

```
14
16
0 1
0 2
0 4
1 3
1 5
2 7
2 8
4 6
4 2
3 2
5 4
6 9
6 10
2 11
10 12
10 13
0
3
```

Stranice do kojih se ne može stići: 12, 13

```
13
19
0 1
0 2
0 3
0 6
1 8
2 4
2 9
3 4
4 9
4 10
4 7
4 5
5 7
6 7
7 10
7 11
8 11
9 10
10 12
0
2
```

Stranice do kojih se ne može stići: 5, 10, 11, 12