

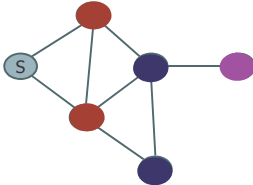
סריקה של גרף

- סריקת גרף פירושה מעבר שיטתי על קשתות הגרף לצורך ביקור בקדקודיו.
- שיטות לסריקת הגרפים הן לב ליבו של תחום האלגוריתמים על גרפים
- אלגוריתם הסורק גרף יכול לגלות דברים רבים על מבנהו

Breadth-First Search
BFS

BFS

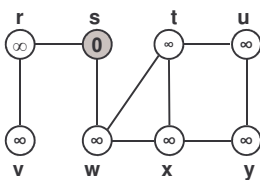
- שלבי עבודה של האלגוריתם:
 - מגלה את כל הקודקודים הנמצאים במרחק 1 מ-s
 - אחר כך את כל הקודקודים הנמצאים במרחק 2 מ-s
 - אחר כך את כל הקודקודים הנמצאים במרחק 3 מ-s
 - ...



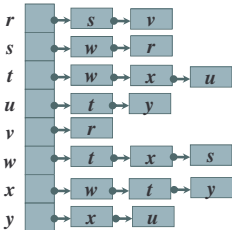
חיפוש לרוחב (breadth-first search)
BFS

- חיפוש לרוחב (BFS) הוא אחד האלגוריתמים הפשוטים לסריקת הגרפים.
- פועל גם על גרפים מכוונים וגם על בלתי מכוונים
- בהינתן גרף $G=(V,E)$ וקדקוד מסוים s משמש כמקור (source), אלגוריתם BFS
- מגלה את כל הקודקודים שניתן להגיע אליהם מ-s
- מחשב מסלול קצר ביותר (מספר המינימאלי של קשתות) מ-s לכל הקודקודים שניתן להגיע אליהם מ-s
- בונה "עץ רחב" ששורשו s

דוגמה

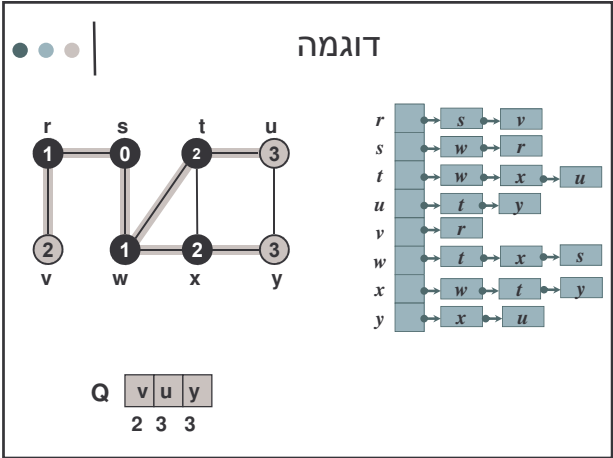
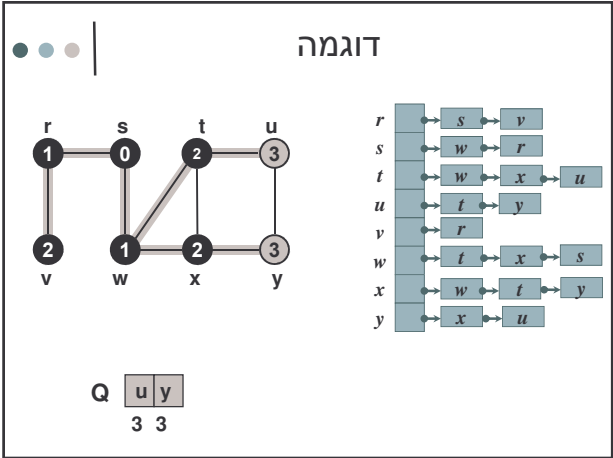
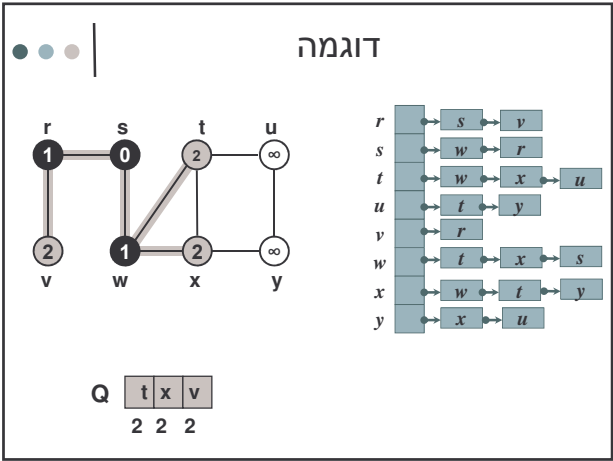
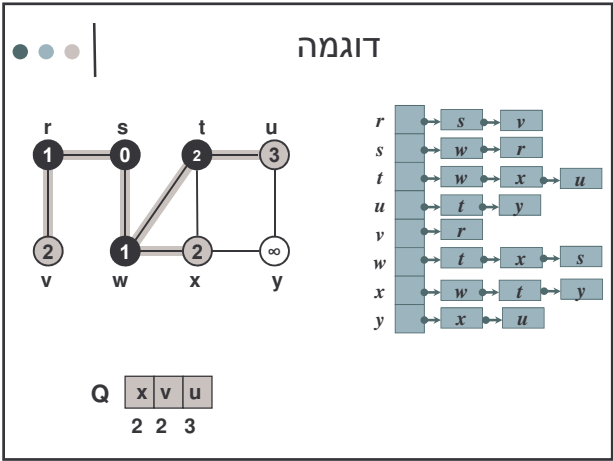
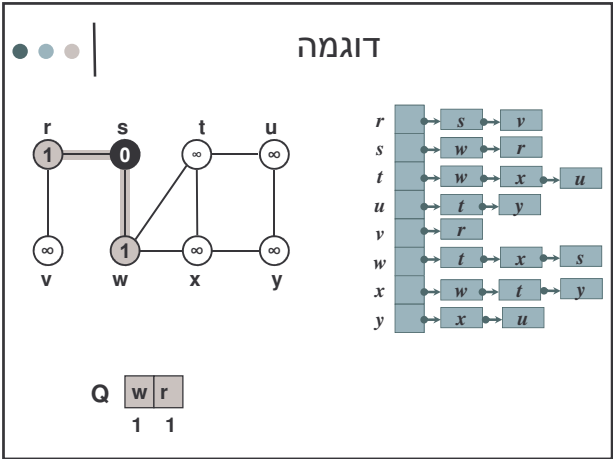
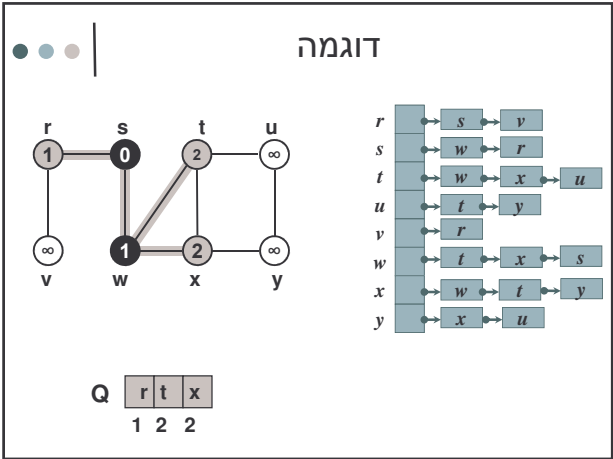


Q s
0



BFS – סיווג קדקודים

- קדקוד שטרם התגלה
- קדקוד שהתגלה אבל לא סיימנו טיפול בו
- קדקוד שהתגלה וסיימנו טיפול בו
- אלגוריתם משתמש בתור FIFO לניהול קבוצת הקודקודים האפורים



דוגמה

r

s

t

u

v

w

x

y

1

0

2

3

2

1

2

3

→

→

→

→

→

→

→

→

s

w

t

r

t

w

x

x

→

→

→

→

→

→

→

→

u

y

s

y

u

Q ∅

דוגמה

r

s

t

u

v

w

x

y

1

0

2

3

2

1

2

3

→

→

→

→

→

→

→

→

s

w

t

r

t

w

x

x

→

→

→

→

→

→

→

→

u

y

s

y

u

Q y
3

מסלולים קצרים ביותר

נגדיר $\delta(s, v)$

- אורך המסלול הקצר ביותר מ-s ל-v (מספר מינימלי של קשתות)
- ∞ אם לא קיים מסלול מ-s ל-v

מסלול באורך $\delta(s, v)$ מ-s ל-v נקרא מסלול קצר ביותר

$\delta(s, v) = 3$

BFS

```
BFS(G=(V, E), s)
// d[u] – distance from s to u
// π[u] – predecessor of u
1 for each vertex u ∈ V - {s}
2   color[u] ← WHITE
3   d[u] ← ∞
4   π[u] ← NULL
5 color[s] ← GRAY
6 d[s] ← 0
7 π[s] ← NULL
8 Q ← ∅
9 ENQUEUE(Q, s)
```

```
10 while Q ≠ ∅
11   u ← DEQUEUE(Q)
12   for each v ∈ Adj[u]
13     if color[v] = WHITE
14       color[v] ← GRAY
15       d[v] ← d[u] + 1
16       π[v] ← u
17   ENQUEUE(Q, v)
18   color[u] ← BLACK
```

זמן ריצה $O(V + E)$

עץ רוחב

נגדיר $G_\pi = (V_\pi, E_\pi)$

תת-גרף הקודמים (predecessor subgraph) של G:

$$V_\pi = \{v \in V \mid \pi[v] \neq \text{NULL}\} \cup \{s\}$$
$$E_\pi = \{(\pi[v], v) \in E \mid v \in V_\pi - \{s\}\}$$

כאשר BFS מופעל על גרף $G = (V, E)$, הוא בונה את השדות π כך שתת-גרף הקודמים $G_\pi = (V_\pi, E_\pi)$ הוא עץ רוחב

מסלולים קצרים ביותר

משפט: בהינתן גרף $G=(V,E)$ וקודקוד מקור s, BFS

- מגלה את כל הקודקודים שניתן להגיע אליהם מ-s
- מחשב מסלול קצר ביותר מ-s לכל הקודקודים שניתן להגיע אליהם מ-s, כלומר $d[v] = \delta(s, v)$ לכל קודקוד v.
- בונה "עץ רוחב", ששורשו s, המכיל את כל הקודקודים שניתן להגיע אליהם מ-s. המסלול מ-s ל-v בעץ הרוחב הוא מסלול באורך $\delta(s, v)$.



מסלול קצר ביותר מ-s ל-v

PRINT-PATH(G, s, v)

```
1 if  $v = s$ 
2   print  $s$ 
3 else if  $\pi[v] = \text{NULL}$ 
4   print "no path from"  $s$  "to"  $v$  "exists"
5 else PRINT-PATH( $G, s, \pi[v]$ )
6   print  $v$ 
```

זמן ריצה ?