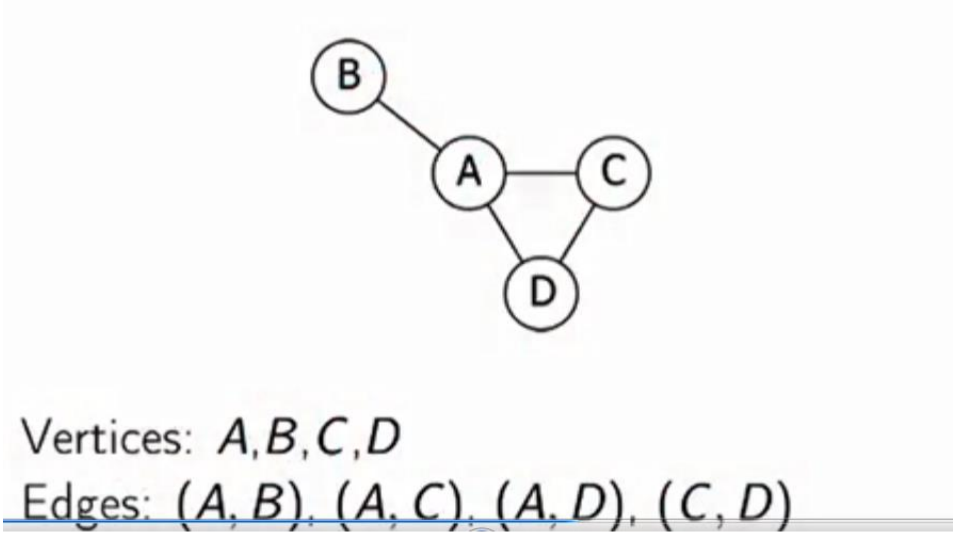
גרפים



הגדרות קצרצרות [לא פורמאליות..]:

קודקוד- איבר המייצג נקודה על הגרף

צלע – חיבור בין שני קודקודים

לולאה – צלע שיוצאת מקודקוד לעצמו

גרף לא מכוון – אין כיוון לצלע

גרף לא מכוון <u>פשוט</u> – אין צלע מקודקוד לעצמו ואין צלעות כפולות מאותו קודקוד א לאותו קודקוד ב.

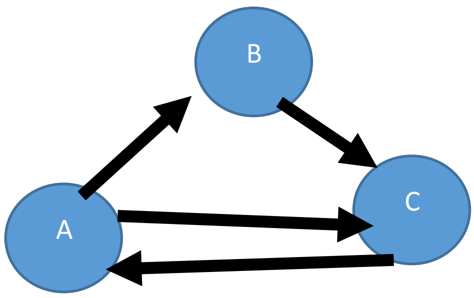
גרף מכוון – יש כיוון לצלע

גרף מכוון <u>פשוט</u> - אין צלע מקודקוד לעצמו ואין צלעות כפולות מאותו קודקוד א לאותו קודקוד ב

גרף ממושקל – לכל צלע יש משקל

גרף לא מכוון קשיר – ניתן להגיע מכל קודקוד לכל קודקוד אחר

גרף מכוון קשיר היטב [או קשיר בחוזקה] – ישנו מסלול מכוון מכל קודקוד לכל קודקוד.



רכיב קשיר בגרף לא מכוון – חלק מהגרף המהווה בפני עצמו גרף קשיר

רכיב קשיר היטב בגרף מכוון - חלק מהגרף המהווה בפני עצמו גרף קשיר היטב.

דרגה של קודקוד [d / deg] – כמות הצלעות היוצאות מהקודקוד בגרף לא מכוון.

בגרף מכוון – דרגת יציאה זה כמות הצלעות היוצאות ודרגת כניסה זה כמות הצלעות הנכנסות.

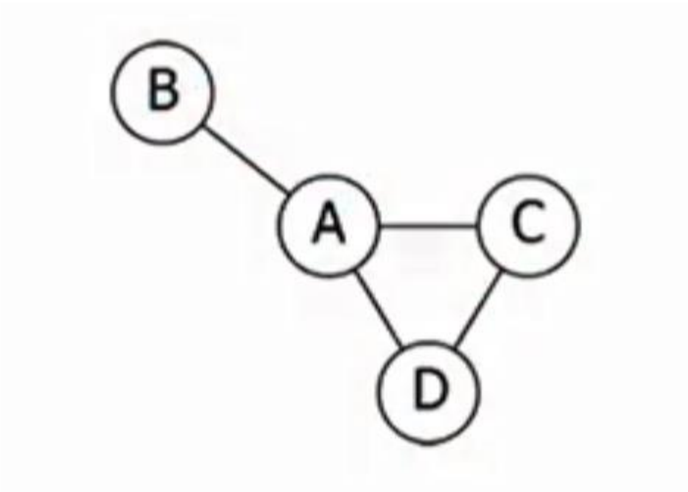
3 דרכים לייצוג גרף במחשב:

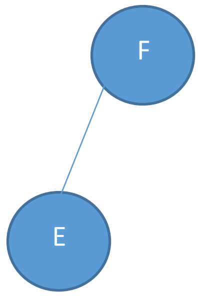
- רשימת צלעות
- מטריצה

	A	B	C	D
A	0	1	1	1
B	1	0	0	0
C	1	0	0	1
D	1	0	1	0

- רשימת שכנויות

לדוגמה:





A : B,C,D
B : A
C : A, D
D: A, C
E: F
F: E

בסיבוכיות נשתמש בעיקר בשני מונחים:

V – קבוצת הקודקודים בגרף

|V| - כמות הקודקודים בגרף

E - רשימת הצלעות בגרף

|E| -כמות הצלעות בגרף

Op.	Is Edge?	List Edge	List Nbrs.
Adj. Matrix	$\Theta(1)$	$\Theta(\|V\|^2)$	$\Theta(\|V\|)$
Edge List	$\Theta(\|E\|)$	$\Theta(\|E\|)$	$\Theta(\|E\|)$
Adj. List	$\Theta(\deg)$	$\Theta(\|E\|)$	$\Theta(\deg)$

סיור בקודקוד בודד לכל הקודקודים הניתנים להגעה ממנו:

```
Explore(v)

visited(v) ← true
for (v, w) ∈ E:
  if not visited(w):
    Explore(w)
```

מעבר לעומק [DFS] על כל הקודקודים בגרף:

```
DFS(G)

for all v ∈ V:     mark v unvisited
for v ∈ V:
  if not visited(v):
    Explore(v)
```

סיבוכיות DFS: O(|E|+|V|)

<u>מציאת רכיבים קשירים בגרף לא מכוון– כל רכיב מקבל מספר משלו cc:</u>

```
Explore(v)
visited(v) ← true
CCnum(v) ← cc
for (v, w) ∈ E:
  if not visited(w):
    Explore(w)
```

```
DFS(G)
for all v ∈ V mark v unvisited
cc ← 1
for v ∈ V:
  if not visited(v):
    Explore(v)
    cc ← cc + 1
```

סיבוכיות: O(|V|+|E|)

<u>מיון טופולוגי [סדר הגיוני של קודקודים בגרף DAG – (גמ"ל- גרף מכוון ללא מעגלים) שמייצג תלויות]:</u>

לכל קודקוד נשמר הPOST שזה מתי סיימנו לטפל בו [בסוף ה explore שמים לקודקוד את הערך post]

```
TopologicalSort(G)
DFS(G)
sort vertices by reverse post-order
```

סיבוכיות: O(|V|+|E|)

<u>מציאת רכיבים קשירים היטב בגרף מכוון</u>

G^R – הגרף המשוחלף, כלומר עם צלעות בכיוון ההפוך

```
SCCs(G)
Run DFS(G^R)
for v ∈ V in reverse postorder:
  if not visited(v):
    Explore(v)
    mark visited vertices
      as new SCC
```

סיבוכיות: O(|V|+|E|)

<u>מציאת המרחק הקטן ביותר מקודקוד S לכל הקודקודים בגרף לא ממושקל – מעבר לרוחב BFS</u>

```
BFS(G, S)
for all u ∈ V:
  dist[u] ← ∞
dist[S] ← 0
Q ← {S} {queue containing just S}
while Q is not empty:
  u ← Dequeue(Q)
  for all (u, v) ∈ E:
    if dist[v] = ∞:
      Enqueue(Q, v)
      dist[v] ← dist[u] + 1
```

סיבוכיות: O(|V|+|E|)

<u>מציאת המסלול הקצר ביותר בגרף מכוון עם משקלות חיוביים בלבד – האלגוריתם של דייקסטרה</u>

$$
\begin{array}{l}
\texttt{Dijkstra}(G,S) \\
\hline
\texttt{for all } u \in V: \\
\quad \texttt{dist}[u] \leftarrow \infty, \texttt{prev}[u] \leftarrow \texttt{nil} \\
\texttt{dist}[S] \leftarrow 0 \\
H \leftarrow \texttt{MakeQueue}(V) \ \texttt{\{dist-values as keys\}} \\
\texttt{while } H \texttt{ is not empty:} \\
\quad u \leftarrow \texttt{ExtractMin}(H) \\
\quad \texttt{for all } (u,v) \in E: \\
\qquad \texttt{if } \ dist[v] > dist[u] + w(u,v): \\
\qquad\quad dist[v] \leftarrow dist[u] + w(u,v) \\
\qquad\quad prev[v] \leftarrow u \\
\qquad\quad \texttt{ChangePriority}(H, v, dist[v])
\end{array}
$$

סיבוכיות: $O((|V|+|E|)\log|V|)$

מציאת המסלול הקצר ביותר בגרף מכוון שמכיל גם משקלות שליליים – האלגוריתם של בלמן ופורד

$$
\begin{array}{l}
\texttt{BellmanFord}(G,S) \\
\hline
\texttt{\{no negative weight cycles in } G\texttt{\}} \\
\texttt{for all } u \in V: \\
\quad \texttt{dist}[u] \leftarrow \infty \\
\quad \texttt{prev}[u] \leftarrow \texttt{nil} \\
dist[S] \leftarrow 0 \\
\texttt{repeat } |V|-1 \texttt{ times:} \\
\quad \texttt{for all } (u,v) \in E: \\
\qquad \texttt{Relax}(u,v)
\end{array}
$$

כדי לזהות מעגל שלילי: עוברים בסוף עוד פעם על הfor שמבצע הפחתות ואם בוצעה הפחתה – יש מעגל שלילי...

סיבוכיות: O(|V||E|)

האלגוריתם של קרוסקל למציאת עץ פורש מינימלי בגרף לא מכוון קשיר

```
Kruskal(G)

for all u ∈ V:
  MakeSet(v)
X ← empty set
sort the edges E by weight
for all {u, v} ∈ E in non-decreasing
  weight order:
  if Find(u) ≠ Find(v):
    add {u, v} to X
    Union(u, v)
return X
```

סיבוכיות: O(|E|log|v|)

<u>האלגוריתם של פרים למציאת הנ"ל</u>

Prim's Algorithm

```
Prim(G)

for all u ∈ V:
  cost[u] ← ∞,  parent[u] ← nil
pick any initial vertex u0
cost[u0] ← 0
PrioQ ← MakeQueue(V)      {priority is cost}
while PrioQ is not empty:
  v ← ExtractMin(PrioQ)
  for all {v,z} ∈ E:
    if z ∈ PrioQ and cost[z] > w(v,z):
      cost[z] ← w(v,z),  parent[z] ← v
      ChangePriority(PrioQ, z, cost[z])
```

סיבוכיות: O(|E|log|V|)