

גרף מכוון קשיר: גרף מכוון G עבורו לכל $u, v \in V(G)$ קיים מסלול מ- u ל- v או מסלול מ- v ל- u .
 גרף מכוון קשיר חזק: גרף מכוון G עבורו לכל $u, v \in V(G)$ קיים מסלול מ- u ל- v .
 אלגוריתם BFS: יהי G גרף ויהי $s \in V(G)$ אזי

```
function BFS( $G, s$ ):
    ( $d, \pi, \text{color}$ )  $\leftarrow$  dict( $V(G)$ )
    for  $u \in V(G) \setminus \{s\}$  do
        color[ $u$ ]  $\leftarrow$  White
        d[ $u$ ]  $\leftarrow$   $\infty$ 
         $\pi[u]$   $\leftarrow$  Null
    end
    color[ $s$ ]  $\leftarrow$  Grey
    d[ $s$ ]  $\leftarrow$  0
     $\pi[s]$   $\leftarrow$  Null
    Q  $\leftarrow$  queue()
    while Q  $\neq \emptyset$  do
         $u \leftarrow$  Q.head
        for  $v \in \text{Neighbor}(u)$  do
            if color[ $v$ ] = White then
                color[ $v$ ]  $\leftarrow$  Grey
                d[ $v$ ]  $\leftarrow$  d[ $u$ ] + 1
                 $\pi[v]$   $\leftarrow$   $u$ 
                Q.enqueue( $v$ )
            end
        end
        Q.dequeue()
        color[ $u$ ]  $\leftarrow$  Black
    end
    return ( $d, \pi, \text{color}$ )
```

טענה: יהי G גרף ויהי $s \in V(G)$ אזי סיבוכיות זמן הריצה של BFS(G, s) הינה $\mathcal{O}(|V| + |E|)$.
 משפט: יהי G גרף ויהי $s \in V$ אזי $[s]_{\rightarrow} = \{v \in V \mid \text{BFS}(G, s).color[v] = \text{Black}\}$.
 סימון: יהי G גרף ויהיו $u, v \in V$ אזי $\delta(v, u) = \min(\{|\text{len}(\sigma)| \mid \sigma \text{ טיול בין } v, u\})$.
 טענה: יהי G גרף ויהיו $v, u, w \in V$ באשר $(w, u) \in E$ אזי $\delta(v, u) \leq \delta(v, w) + 1$.
 למה: יהי G גרף ויהיו $s, v \in V$ אזי בכל שלב בהרצת BFS(G, s) מתקיים $d[v] \geq \delta(v)$.
 למה: יהי G גרף יהי שלב בהרצת BFS(G, s) בו $Q = (v_1 \dots v_n)$ אזי מתקיים $d[v_i] \leq d[v_{i+1}] + 1$ וכן $d[v_i] \leq d[v_1] + 1$.
 משפט נכונות מרחקים: יהי G גרף ויהיו $s, v \in V$ אזי $\text{BFS}(G, s).d[v] = \delta(v, s)$.
 עץ BFS: יהי G גרף ויהי $s \in V$ נגדיר $V_\pi = \{v \in V \mid \text{BFS}(G, s).\pi[v] \neq \text{Null}\} \cup \{s\}$ וכן $E_\pi = \{(\pi[v], v) \mid v \in V_\pi \setminus \{s\}\}$ אזי $G_\pi = (V_\pi, E_\pi)$.

טענה: יהי G גרף ויהי $s \in V$ אזי

- מתקיים $\deg_{G_\pi}^-(s) = 0$.
- לכל $v \in V(G_\pi)$ מתקיים $\deg_{G_\pi}^-(v) = 1$.
- לכל $v \in V(G_\pi)$ קיים מסלול ב- G_π בין s, v .
- G_π הינו עץ.
- יהי $v \in V(G_\pi)$ ויהי σ מסלול ב- G_π בין s, v אזי σ המסלול הקצר ביותר בין s, v ב- G .

מסלול אוילר: מסלול העובר על כל הקשתות בגרף.

מעגל אוילר: מסלול אוילר שהינו מעגל.

טענה: יהי G גרף קשיר ולא מכוון אזי (יש מעגל אוילר ב- G) $\iff (\deg(v) \in \mathbb{N}_{\text{even}} \text{ מתקיים } v \in V)$
 אלגוריתם למציאת מעגל אוילר: יהי G גרף קשיר ולא מכוון עבורו לכל $u \in V$ מתקיים $\deg(u) \in \mathbb{N}_{\text{even}}$ אזי

```

function EulerCircle( $G, v$ ):
     $\sigma \leftarrow \text{List}(E(G))$ 
     $u \leftarrow \text{Neighbor}(v)$ 
    while  $u \neq v$  do
         $\sigma.append(\{v, u\})$ 
         $G = G \setminus \{\{v, u\}\}$ 
         $u \leftarrow \text{Neighbor}(u)$ 
    end
    if  $\text{length}(\sigma) = |E(G)|$  then
        return  $\sigma$ 
    end
    else
         $w \leftarrow \{x \in V(G) \mid (\exists y \in V(G)).(x, y) \in \sigma \wedge (\deg(x) > 0)\}$ 
         $\sigma[w] = \text{EulerCircle}(G, w)$ 
    end
    return  $\sigma$ 

```

טענה: יהי G גרף קשיר ולא מכוון עבורו לכל $u \in V$ מתקיים $\deg(u) \in \mathbb{N}_{\text{even}}$ ויהי $v \in V(G)$ אזי סיבוכיות זמן הריצה של $\text{EulerCircle}(G, v)$ הינה $\mathcal{O}(|E|)$.

טענה: באלגוריתם EulerCircle כל עוד לולאת ה-while פעילה מתקיים $|\text{Neighbor}(u)| \neq \emptyset$.

משפט: יהי G גרף קשיר ולא מכוון עבורו לכל $v \in V$ מתקיים $\deg(v) \in \mathbb{N}_{\text{even}}$ אזי $\text{EulerCircle}(G)$ הינו מעגל אוילר.

טענה: יהי G גרף קשיר ולא מכוון אזי יש מסלול אוילר שאינו מעגל ב- G $\iff |\{v \in V(G) \mid \deg(v) \in \mathbb{N}_{\text{odd}}\}| = 2$.

אלגוריתם למציאת מסלול אוילר: יהי G גרף קשיר ולא מכוון עבורו $|\{v \in V(G) \mid \deg(v) \in \mathbb{N}_{\text{odd}}\}| = 2$ אזי

```

function EulerPath( $G$ ):
     $\{v, u\} \leftarrow \{v \in V(G) \mid \deg(v) \in \mathbb{N}_{\text{odd}}\}$ 
     $G = G + \{\{v, u\}\}$ 
     $\sigma = \text{EulerCircle}(G, v)$ 
    return  $\sigma \setminus \{v, u\}$ 

```

טענה: יהי G גרף לא מכוון אזי G דו-צדדי \iff (לא קיים ב- G מעגל באורך אי-זוגי).

אלגוריתם זיהוי גרפים דו-צדדיים: יהי G גרף לא מכוון ופשוט אזי

```

function IsBipartite( $G$ ):
     $(d, \pi, \text{color}) \leftarrow \text{BFS}(G)$ 
    for  $(v, u) \in V$  do
        if  $d(v) = d(u)$  then
            return false
        end
    end
    return true

```

טענה: יהי G גרף לא מכוון ופשוט אזי G דו צדדי $\iff (\text{IsBipartite}(G) = \text{true})$.

מסלול קצר ביותר בין קודקודים: יהי G גרף ויהיו $s, t \in V$ אזי מסלול σ מ- s ל- t עבורו $|\sigma| = \min\{|\tau| \mid \tau \text{ מסלול } s\text{-}t\}$.

גרף המסלולים הקצרים ביותר מקודקוד (גרף מק"ב): יהי G גרף ויהי $s \in V$ נגדיר

$E' = \{e \in E \mid s \text{ מיוצא מ-} e\}$ אזי (V, E') .

אלגוריתם למציאת גרף המסלולים הקצרים ביותר מקודקוד: יהי G גרף ויהי $s \in V$ אזי

```

function ShortestPathGraph( $G, s$ ):
    ( $d, \pi, \text{color}$ )  $\leftarrow$  BFS( $G$ )
     $E' \leftarrow E(G_\pi)$ 
    for ( $u, v$ )  $\in E(G)$  do
        if  $|height_{G_\pi}(u) - height_{G_\pi}(v)| = 1$  then
             $E'.\text{append}((u, v))$ 
        end
    end
    return ( $V(G), E'$ )

```

טענה: תהא $e \in E$ אזי e מחברת בין רמות עוקבות ביער BFS G_π $\iff e$ קשת במק"ב.

מסקנה: יהי G גרף ויהי $s \in V$ אזי ShortestPathGraph(G, s) הינו גרף מק"ב מ- s .

גרף המסלולים הקצרים ביותר בין קודקודים: יהי G גרף ויהיו $s, t \in V$ נגדיר

$E' = \{e \in E \mid t \text{ מ-} s \text{ היוצא ביותר}\}$ אזי (V, E') .

טענה: יהי G גרף מכוון ויהיו $s, t \in V$ אזי קיים אלגוריתם לחישוב גרף המסלולים הקצרים ביותר מ- s ל- t בסיבוכיות $\mathcal{O}(|V| + |E|)$.

אלגוריתם DFS: יהי G גרף ויהי $s \in V$ אזי

```

function DFS( $G, s$ ):
    ( $k, \pi$ )  $\leftarrow$  dict( $V$ )
    color  $\leftarrow$  dict( $E$ )
     $k[s] \leftarrow 1$ 
     $\pi[s] \leftarrow \text{Null}$ 
    for  $u \in V \setminus s$  do
         $k[u] \leftarrow 0$ 
         $\pi[u] \leftarrow \text{Null}$ 
    end
    for  $e \in E$  do
        color[e]  $\leftarrow$  White
    end
     $i \leftarrow 2$ 
     $v \leftarrow s$ 
    while ( $\exists u \in \text{Adj}(v). \text{color}[(v, u)] = \text{White} \vee (\pi[v] \neq \text{Null})$ ) do
        if  $\{u \in \text{Adj}(v) \mid \text{color}[(v, u)] = \text{White}\} \neq \emptyset$  then
             $w \leftarrow \{u \in \text{Adj}(v) \mid \text{color}[(v, u)] = \text{White}\}$ 
            color[( $v, w$ )]  $\leftarrow$  Black
            if  $k[w] = 0$  then
                 $k[w] \leftarrow i$ 
                 $\pi[w] \leftarrow v$ 
                 $v \leftarrow w$ 
                 $i \leftarrow i + 1$ 
            else
                 $v \leftarrow \pi[v]$ 
            end
        end
    end
    return ( $k, \pi$ )

```

טענה: יהי G גרף ויהי $s \in V$ אזי סיבוכיות זמן הריצה של DFS(G, s) הינה $\mathcal{O}(|V| + |E|)$.

זמן גילוי: יהי G גרף ויהי $s \in V$ אזי k בהרצת DFS(G, s).

טענה: יהי G גרף ויהיו $s, v \in V$ באשר $s \rightarrow v$ אזי בהרצת DFS(G, s) מתקיים $k[v] > 0$.

עץ DFS: יהי G גרף ויהי $s \in V$ נגדיר $V_\pi = \{v \in V \mid \text{DFS}(G, s). \pi[v] \neq \text{Null}\} \cup \{s\}$ וכן $E_\pi = \{(\pi[v], v) \mid v \in V_\pi \setminus \{s\}\}$ אזי $G_\pi = (V_\pi, E_\pi)$.

טענה: עץ DFS הינו עץ.

קשתות ביחס לריצת DFS: יהי G גרף ויהי G_π יער DFS אזי

• קשתות עץ: קשת $e \in E(G)$ עברה $e \in E(G_\pi)$.

- קשתות קדמיות: קשת $(u, v) \in E(G)$ עבורה $(u, v) \notin E(G_\pi)$ וכן u הינו אב של v .
 - קשתות אחוריות: קשת $(u, v) \in E(G)$ עבורה $(u, v) \notin E(G_\pi)$ וכן v הינו אב של u .
 - קשתות חוצות: קשת $e \in E(G)$ שאינה קשת עץ או קדמית או אחורית.
- טענה:** יהי G גרף לא מכוון ותהא (u, v) קשת עץ אזי u צאצא של v בגרף G_π או v צאצא של u בגרף G_π .
- מסקנה:** יהי G גרף לא מכוון אזי לא קיימות קשתות חוצות.
- אלגוריתם DFS בעל זמני נסיגה:** יהי G גרף אזי

```
function DFS(G):
    (k, f, π, color, low) ← dict(V)
    for u ∈ V do
        k[u] ← 0
        π[u] ← Null
        color ← White
        low ← ∞
    end
    i ← 0
    for s ∈ V do
        if k[s] = 0 then
            DFS-VISIT(s, k, f, π, i)
        end
    end
    return (k, f, π, low)
```

```
function DFS-VISIT(v, k, f, π, color, low, i):
    color[u] ← Gray
    i ← i + 1
    k[v] ← i
    for w ∈ Adj(v) do
        if (color[v] = Gray) ∧ (v ≠ π[u]) then
            low ← min(low[u], k[v])
        else if color[v] = White then
            π[w] ← v
            DFS-VISIT(w, k, f, π, color, low, i)
            low ← min(low[u], low[v])
        end
    end
    color[u] ← Black
    i ← i + 1
    f[v] ← i
```

- זמן נסיגה:** יהי G גרף ויהי $s \in V(G)$ אזי f בהרצת $\text{DFS}(G)$.
- טענה Gray Path Lemma:** יהיו $v, u \in V$ אזי v צאצא של u ביער G_π $\iff (k[u] < k[v] < f[u])$.
- טענה:** יהיו $v, u \in V$ אזי (u, v) קשת חוצה $\iff (f[v] < k[u])$.
- משפט הסוגריים:** יהי G גרף ויהיו $u, v \in V$ אזי מתקיים בדיוק אחד מהבאים
- מתקיים $[k(u), f(u)] \cap [k(v), f(v)] = \emptyset$ וכן u, v אינם צאצא־אב ביער G_π .
 - מתקיים $[k(u), f(u)] \subset [k(v), f(v)]$ וכן u צאצא של v ביער G_π .
 - מתקיים $[k(u), f(u)] \supset [k(v), f(v)]$ וכן v צאצא של u ביער G_π .
- משפט המסלול הלבן:** יהי G גרף ויהיו $u, v \in V$ אזי u, v צאצא של u ביער G_π \iff (בזמן $k(u)$ באלגוריתם $\text{DFS}(G)$ יש מסלול לבן מ־ u ל־ v).
- גרף מכוון אציקלי:** גרף מכוון G בו לא קיים מעגל.
- מיון טופולוגי:** יהי G גרף מכוון אזי יחס סדר \prec על V המקיים לכל $u, v \in V$ אם $(u, v) \in E$ אזי $u \prec v$.
- משפט:** יהי G גרף מכוון אזי $(G$ אציקלי) \iff (קיים מיון טופולוגי על G).

טענה אלגוריתם קנות: יהי G גרף מכוון אזי קיים אלגוריתם לבדיקת קיום מיון טופולוגי בסיבוכיות $\mathcal{O}(|V| + |E|)$.

משפט: יהי G גרף מכוון אזי $(G \text{ רציקלי}) \iff (\text{אין קשתות אחוריות ב-} G)$.

טענה: יהי G גרף מכוון אציקלי אזי f המתקבלת מהרצת $\text{DFS}(G)$ משרה מיון טופולוגי על G .

קודקוד מנתק: יהי G גרף מכוון אזי $v \in V(G)$ עבורו $\left| G - \frac{v}{G - \{v\}} \right| \leq \left| G - \frac{v}{G} \right|$.

אב חורג: יהי G גרף מכוון ויהי $v \in V$ אזי $w \in V$ עבורו (w, v) קשת אחורית.

זמן גילוי האב החורג המוקדם ביותר: יהי G גרף אזי low בהרצת $\text{DFS}(G)$.

אלגוריתם למציאת כל הקודקודים המנתקים: יהי G גרף מכוון וקשיר אזי

```
function DetachableVertices(G):
    s ← V
    (k, f, π, low) ← DFS(G, s)
    A ← set(V)
    if |AdjGπ(s)| ≠ 1 then
        A.append(s)
    end
    for u ∈ V \ {s} do
        if ∃v ∈ children(u).low[v] ≥ k[u] then
            A.append(u)
        end
    end
    return A
```

טענה: יהי G גרף מכוון וקשיר אזי סיבוכיות $\text{DetachableVertices}(G)$ הינה $\mathcal{O}(|V| + |E|)$.

טענה: יהי G גרף מכוון וקשיר אזי $\text{DetachableVertices}(G)$ הינה קבוצת כל הקודקודים המנתקים.

רכיב קשיר היטב (רק"ה): יהי G גרף מכוון אזי קבוצה $C \subseteq V$ מקסימלית בגודלה עבורה לכל $u, v \in C$ קיים מסלול מ- u ל- v וכן מ- v ל- u .

גרף הופכי/משוחלף: יהי G גרף מכוון נגדיר $E' = \{(v, u) \mid (u, v) \in E\}$ אזי $G^T = (V, E')$.

טענה: יהי G גרף מכוון ותהא $C \subseteq V$ אזי $(C \text{ רק"ה של } G) \iff (C \text{ רק"ה של } G^T)$.

אלגוריתם קוסראט'ו-שריר למציאת רכיבים קשירים היטב: יהי G גרף מכוון אזי

```
function SCC(G):
    (k, f, π) ← DFS(G)
    /* In the next line, in the main loop of DFS we go through the vertices in decreasing order of f[u] */
    (k', f', π') ← DFS(GT)
    A ← set(set(V))
    for v ∈ V do
        A.append([v] →GTπ')
    end
    return A
```

גרף הרכיבים: יהי G גרף מכוון נגדיר $E^* = \{(A, B) \in \text{SCC}(G)^2 \mid \exists u \in A. \exists v \in B. (u, v) \in E\}$ אזי $G^* = (\text{SCC}(G), E^*)$ **אלגוריתם למציאת גרף הרכיבים:** יהי G גרף מכוון אזי

```

function KosarajuSharir(G):
    V* ← SCC(G)
    E* ← set((V*)2)
    for (u, v) ∈ E do
        if  $[v] \xrightarrow{G_\pi^T} \neq [u] \xrightarrow{G_\pi^T}$  then
            E*.append( $\left(\left([v] \xrightarrow{G_\pi^T}, [u] \xrightarrow{G_\pi^T}\right)\right)$ )
        end
    end
    end
    return (V*, E*)

```

למה: יהי G גרף מכוון אזי G^* אציקלי.

הגדרה: יהי G גרף ותהא $U \subseteq V$

• זמן גילוי: $k(U) = \min_{u \in U} (k[u])$

• זמן נסיגה: $f(U) = \max_{u \in U} (f[u])$

למה: יהי G גרף מכוון יהיו $C_1, C_2 \subseteq V$ רק"ה באשר $(C_1, C_2) \in E(G^*)$ אזי $f(C_2) < f(C_1)$

מסקנה: יהי G גרף מכוון יהיו $C_1, C_2 \subseteq V$ רק"ה באשר $(C_1, C_2) \in E((G^T)^*)$ אזי $f(C_2) > f(C_1)$

משפט: יהי G גרף מכוון ויהי $C \subseteq V$ אזי C רק"ה $\iff (C \in \text{SCC}(G))$

מסקנה: יהי G גרף מכוון אזי $G^* = \text{KosarajuSharir}(G)$

קבוצת מוצא: יהי G גרף מכוון אזי $S \subseteq V$ המקיימת $s \rightarrow v$ $\forall s \in S, v \in V$

אלגוריתם למציאת קבוצת מוצא מינימלית: יהי G גרף מכוון אזי

```

function MinimalOriginSet(G):
    A ← set(V(G))
    G* ← ComponentGraph(G)
    for C ∈ V(G*) do
        v ← {u ∈ C |  $\nexists w \in V(G) \setminus C. (w, u) \in E(G)$ }
        A.append(v)
    end
    return A

```

טענה: יהי G גרף מכוון אזי $\text{MinimalOriginSet}(G)$ קבוצת מוצא מינימלית.

טענה: יהי G גרף מכוון וקשיר אזי סיבוכיות $\text{MinimalOriginSet}(G)$ הינה $\mathcal{O}(|V| + |E|)$

טענה: יהי G גרף מכוון ותהא $S \subseteq V$ אזי קיים אלגוריתם הבודק האם קיים הילוך σ העובר על S בסיבוכיות $\mathcal{O}(|V| + |E|)$

גרף ממושקל: יהי G גרף ותהא $w : E \rightarrow \mathbb{R}$ אזי (G, w)

עץ פורש: יהי G גרף קשיר לא מכוון אזי תת־גרף $T \leq G$ באשר T עץ וכן $V(T) = V(G)$

משקל עץ פורש: יהי G גרף קשיר לא מכוון ויהי $T \leq G$ עץ פורש אזי $w(T) = \sum_{e \in E(T)} w(e)$

עץ פורש מינימלי (עפ"מ): יהי G גרף קשיר לא מכוון אזי עץ פורש S עבורו $T \leq G$ $w(T) = \min \{w(S) \mid S \text{ עץ פורש של } G\}$

חתך: יהי G גרף אזי $A, B \subseteq V(G)$ עבורם $A \oplus B = V(G)$

קשתות החתך/חוצות: יהי G גרף ויהי $A, B \subseteq V(G)$ חתך אזי $\{(u, v) \in E(G) \mid (u \in A) \wedge (v \in B)\}$

טענה: יהי $T \leq G$ עץ פורש ותהא $e \in E(G) \setminus E(T)$ אזי $T + \{e\}$ בעל מעגל יחיד.

טענה: יהי $T \leq G$ עץ פורש תהא $e_1 \in E(G) \setminus E(T)$ ותהא $e_2 \in E(T + \{e_1\})$ אשר הינה חלק ממעגל אזי $T + \{e_1\} - \{e_2\}$ עץ פורש.

טענה: יהי $T \leq G$ עץ פורש ותהא $e \in E(T)$ אזי $T - \{e\}$ הינו יער בעל שני עצים.

מסקנה: יהי $T \leq G$ עץ פורש תהא $e \in E(T)$ ויהי $v \in V(G)$ אזי $[v] \xrightarrow{T - \{e\}} V(G) \setminus [v] \xrightarrow{T - \{e\}}$ חתך של G

אלגוריתם גנרי למציאת עץ פורש מינימלי: יהי G גרף קשיר לא מכוון וממושקל אזי

```

function MST( $G, w$ ):
    color  $\leftarrow$  dict( $E$ )
    for  $e \in E$  do
        | color[ $e$ ] = White
    end
    while  $\exists e \in E$ .color[ $e$ ] = White do
        Blueless  $\leftarrow \{A \subseteq V \mid \forall e \in (A^2 \cap E).color[e] \neq \text{Blue}\}$ 
        Redless  $\leftarrow \{\sigma \text{ circle in } G \mid \forall i \in [\text{len}(\sigma)].color[(\sigma[i], \sigma[i+1])] \neq \text{Red}\}$ 
        if Blueless  $\neq \emptyset$  then
            |  $A \leftarrow$  Blueless
            |  $f \leftarrow \text{argmin}_{e \in A^2 \cap E}(w(e))$ 
            | color[ $f$ ] = Blue
        end
        if Redless  $\neq \emptyset$  then
            |  $\sigma \leftarrow$  Redless
            |  $f \leftarrow \text{argmax}_{e \in \sigma}(w(e))$ 
            | color[ $f$ ] = Red
        end
    end
    return ( $V, \{e \in E \mid \text{color}[e] = \text{Blue}\}$ )

```

טענה: יהי G גרף קשיר לא מכוון וממושקל w ותהא $a \in E$ עבורה $\text{color}[a] = \text{White}$ באיטרציה של $\text{MST}(G)$ אזי קיימת $e \in E$ אשר ניתנת לצביעה.

מסקנה: יהי G גרף קשיר לא מכוון וממושקל w אזי $\text{MST}(G)$ צובעת $|E|$ קשתות.

טענה: יהי G גרף קשיר לא מכוון וממושקל w אזי בכל איטרציה של $\text{MST}(G)$ קיים $T \leq G$ עפ"מ עבורו

- לכל $e \in E(T)$ המקיימת $\text{color}[e] = \text{Blue}$ מתקיים
- לכל $e \in E$ המקיימת $\text{color}[e] = \text{Red}$ מתקיים $e \notin E(T)$

מסקנה: יהי G גרף קשיר לא מכוון וממושקל w אזי $\text{MST}(G)$ עפ"מ של G .

אלגוריתם פרים למציאת עץ פורש מינימלי: יהי G גרף קשיר לא מכוון וממושקל w אזי

```

function Prim'sAlgorithm( $G$ ):
    color  $\leftarrow$  dict( $E$ )
     $U \leftarrow \text{set}(V)$ 
    for  $e \in E$  do
        | color[ $e$ ] = White
    end
     $r \leftarrow V$ 
     $U.append(r)$ 
    while  $U \neq V$  do
        ( $u, v$ )  $\leftarrow \text{argmin}_{e \in U \times (V \setminus U)}(w(e))$ 
        color[ $(u, v)$ ] = Blue
         $U.append(v)$ 
        for  $w \in U$  do
            if ( $w, v$ )  $\in E$  then
                | color[ $(w, v)$ ] = Red
            end
        end
    end
    return ( $V, \{e \in E \mid \text{color}[e] = \text{Blue}\}$ )

```

טענה: יהי G גרף קשיר לא מכוון וממושקל w אזי כל צביעת קשת באלגוריתם $\text{Prim'sAlgorithm}(G)$ נעשית כמו באלגוריתם הגנרי.

מסקנה: יהי G גרף קשיר לא מכוון וממושקל w אזי $\text{Prim'sAlgorithm}(G)$ עפ"מ של G .

משפט: יהי G גרף קשיר לא מכוון וממושקל w אזי ניתן לממש את $\text{Prim'sAlgorithm}(G)$ עם ערימת מינימום בסיבוכיות $\mathcal{O}(|E| \log |V|)$.

הערה: יהי G גרף קשיר לא מכוון וממושקל w אזי ניתן לממש את Prim's Algorithm (G) בסיבוכיות $\mathcal{O}(|E| + |V| \log |V|)$.
אלגוריתם קרוסקל למציאת עץ פורש מינימלי: יהי G גרף קשיר לא מכוון וממושקל w אזי

```
function Kruskal'sAlgorithm( $G$ ):
    color  $\leftarrow$  dict( $E$ )
     $L \leftarrow$  sort( $E$ )
    for  $(u, v) \in L$  do
        if  $\exists \sigma \in \{u \rightarrow v\}. \forall i \in [n]. \text{color}(\sigma(i)) = \text{Blue}$  then
            | color[ $e$ ] = Red
        end
        else
            | color[ $e$ ] = Blue
        end
    end
    return ( $V, \{e \in E \mid \text{color}[e] = \text{Blue}\}$ )
```

טענה: יהי G גרף קשיר לא מכוון וממושקל w אזי כל צביעת קשת באלגוריתם Kruskal's Algorithm (G) נעשית כמו באלגוריתם הגנרי.
מסקנה: יהי G גרף קשיר לא מכוון וממושקל w אזי Kruskal's Algorithm (G) ע"מ של G .

משפט: יהי G גרף קשיר לא מכוון וממושקל w אזי ניתן לממש את Kruskal's Algorithm (G) עם Union-Find בסיבוכיות $\mathcal{O}(|E| \log |V|)$ וכן סיבוכיות amortized $\mathcal{O}(|E| \cdot \alpha(|V|))$.
אלגוריתם Borůvka למציאת עץ פורש מינימלי: יהי G גרף קשיר לא מכוון וממושקל w באשר w חח"ע אזי

```
function Borůvka'sAlgorithm( $G$ ):
    Trees  $\leftarrow$  set(set( $G$ ))
    for  $v \in V$  do
        | Trees.append( $\{v\}$ )
    end
    while |Trees|  $\neq 1$  do
        for  $T \in \text{Tree}$  do
            ( $u, v$ )  $\leftarrow$  argmin $_{(u,v) \in V(T) \times V(G)} (w((u, v)))$ 
             $S \leftarrow \{S \in \text{Tree} \mid u \in V(S)\}$ 
             $S \leftarrow S + T + \{(u, v)\}$ 
            Trees.Remove( $T$ )
        end
    end
    A  $\leftarrow$  Trees
    return A
```

טענה: יהי G גרף קשיר לא מכוון וממושקל w באשר w חח"ע אזי סיבוכיות Borůvka's Algorithm (G) הינה $\mathcal{O}(|E| \log |V|)$.
משפט: יהי G גרף קשיר לא מכוון וממושקל w באשר w חח"ע אזי קיים ויחיד $T \leq G$ ע"מ.

משפט: יהי G גרף קשיר לא מכוון וממושקל w באשר w חח"ע אזי Borůvka's Algorithm (G) ע"מ של G .

משפט: יהי G גרף קשיר לא מכוון וממושקל w תהא $A \subseteq E$ יהי C מעגל ותהא $e \in E$ בעלת משקל מקסימלי אזי קיים ע"מ $T \leq G$ עבורו $A \subseteq E(T)$ וכן $e \notin E(T)$.

טענה: יהיו $T_1, T_2 \leq G$ ע"מ ויהיו $\alpha_1 \leq \dots \leq \alpha_n$ ו- $\beta_1 \leq \dots \leq \beta_m$ משקליי הקשתות כולל כפילויות אזי $n = m$ וכן $\alpha_i = \beta_i$ לכל $i \in [n]$.

אלגוריתם למציאת עץ פורש מינימלי המכיל מספר מקסימלי של קשתות מקבוצה מסוימת: יהי G גרף קשיר לא מכוון וממושקל w ותהא $F \subseteq E$ אזי


```

function PrioritizeMST( $G, w, F$ ):
     $\varepsilon \leftarrow \min(\{w(e_1) - w(e_2) \mid (e_1, e_2) \in E \wedge (w(e_1) \neq w(e_2))\})/2$ 
    for  $e \in E$  do
        if  $e \in F$  then
            end
        else
             $w'(e) \leftarrow w(e)$ 
        end
    end
    return Kruskal'sAlgorithm( $G, w'$ )

```

טענה: תהא $F \subseteq E$ ויהי T עפ"מ ביחס ל- w' באלגוריתם PrioritizeMST אזי T עפ"מ ביחס ל- w .

מסקנה: תהא $F \subseteq E$ אזי PrioritizeMST(G, w) עפ"מ ב- G ביחס ל- w .

בעיית שיבוץ המשימות: יהיו $s_1 \dots s_n, f_1 \dots f_n \in \mathbb{R}$ באשר $s_i < f_i$ לכל $i \in [n]$ אזי $\max \{|A| \mid (A \subseteq \{[s_1, f_i]\}_{i=1}^n) \wedge (\forall I, J \in A. I \cap J = \emptyset)\}$

אלגוריתם חמדן לבעיית שיבוץ המשימות: יהיו $s_1 \dots s_n, f_1 \dots f_n \in \mathbb{R}$ באשר $s_i < f_i$ לכל $i \in [n]$ אזי

```

function ActivitySelectionProblem( $s_1, \dots, s_n, f_1, \dots, f_n$ ):
     $F \leftarrow \text{list}([s_1, f_1], \dots, [s_n, f_n])$ 
    /* In the next line, we put all the parameters in F and sort it based on  $f_i$  */
     $F \leftarrow \text{sort}(\{f_1, \dots, f_n\})$ 
     $X \leftarrow \text{list}([s_1, f_1], \dots, [s_n, f_n])$ 
     $X \leftarrow \emptyset$ 
    for  $k \in [1, \dots, n]$  do
        if  $X = \emptyset$  then
             $X.append(L[k])$ 
        else if  $L[k] \cap X.last = \emptyset$  then
             $X.append(L[k])$ 
        end
    end
    return  $X$ 

```

טענה: יהיו $s_1 \dots s_n, f_1 \dots f_n \in \mathbb{R}$ באשר $s_i < f_i$ אזי סיבוכיות ActivitySelectionProblem הינה $O(n \log(n))$.

משפט: לכל $k \in [n]$ באיטרציה ה- k בלולאה ב-ActivitySelectionProblem קיים פתרון לבעיה X^* עבורו $([s_i, f_i] \in X^*) \iff ([s_i, f_i] \in X)$.

מסקנה: יהיו $s_1 \dots s_n, f_1 \dots f_n \in \mathbb{R}$ באשר $s_i < f_i$ אזי ActivitySelectionProblem פתרון לבעיית שיבוץ המשימות.

הערה: כאשר משקל הגרף הוא ℓ הכוונה היא "אורך הקשת" ובכך אנו "מכלילים" גרף ללא משקל בתור גרף עבורו $\ell = 1$.

מעגל שלילי: יהי G מעגל ממושקל ℓ אזי מעגל C עבורו $\ell(C) < 0$.

מסלול קצר ביותר בין קודקודים: יהי G גרף ממושקל ℓ ויהיו $s, t \in V$ אזי מסלול σ מ- s ל- t עבורו $\ell(\sigma) = \min \{\ell(\tau) \mid \tau \in \{s \rightarrow t\}\}$.

למה: יהיו $s, t \in V$ עבורם קיים מסלול מ- s ל- t וכן כל מסלול מ- s ל- t לא עובר דרך קשת השייכת למעגל שלילי אזי קיים מסלול פשוט קצר ביותר בין s ל- t .

למה: יהיו $s, t \in V$ עבורם קיים מסלול מ- s ל- t וכן קיים מסלול מ- s ל- t העובר דרך קשת השייכת למעגל שלילי אזי לא קיים מסלול פשוט קצר ביותר בין s ל- t .

סימון: יהיו $s, t \in V$ אזי $\delta(s, t) = \inf_{\sigma \in \{s \rightarrow t\}} \ell(\sigma)$

בעיית המסלולים הקצרים מנוצא (SSSP): יהי G גרף מכוון ממושקל ℓ ויהי $s \in V$ אזי $T \leq G$ עץ פורש בו כל מסלול מ- s ל- v הינו מסלול קצר ביותר ב- G .

למה אי-שיויון המשולש: יהיו $u, v, w \in V$ אזי $\delta(u, v) \leq \delta(u, w) + \delta(w, v)$

למה תת-מסלול קצר ביותר: יהי σ מסלול קצר ביותר אזי $(\sigma[i], \dots, \sigma[i+k])$ מסלול קצר ביותר.

אלגוריתם בלמן-פורד למציאת מסלולים קצרים ביותר מנוקודת מוצא: יהי G גרף מכוון וממושקל ℓ ויהי $s \in V$ אזי

```

function BellmanFord( $G, \ell, s$ ):
    ( $d, \pi$ )  $\leftarrow$  dict( $V$ )
     $d[s] \leftarrow 0$ 
    for  $u \in V$  do
         $d[u] \leftarrow \infty$ 
         $\pi[u] \leftarrow \text{None}$ 
    end
    ( $c, i$ )  $\leftarrow 1$ 
    while ( $i \leq |V| \wedge (c \neq 0)$ ) do
         $c \leftarrow 0$ 
        for ( $u, v$ )  $\in E$  do
             $c \leftarrow \text{Relax}(\ell, d, u, v)$ 
        end
        end
    end
    return  $c$ 

```

```

function Relax( $\ell, d, u, v$ ):
    if  $d[v] > d[u] + \ell(u, v)$  then
         $d[v] \leftarrow d[u] + \ell(u, v)$ 
         $\pi[v] \leftarrow u$ 
        return 1
    end
    return 0

```

למה: יהיו $s, u, v \in V$ באשר $(u, v) \in E$ וכן בריצת BellmanFord מתקיים $\delta(s, u) \leq d[u]$ אזי $\delta(s, v) \leq d[u] + \ell(u, v)$

מסקנה: יהיו $s, u, v \in V$ באשר $(u, v) \in E$ וכן בריצת BellmanFord מתקיים $\delta(s, u) \leq d[u]$ וכן $\delta(s, v) \leq d[v]$ אזי לאחר הרצת Relax(u, v) מתקיים $\delta(s, v) \leq d[v]$

למה: יהי $s \in V$ עבורו לכל $v \in V$ בריצת BellmanFord מתקיים $\delta(s, v) \leq d[v]$ אזי לאחר כל רצף פעולות Relax נקבל כי לכל $v \in V$ מתקיים $\delta(s, v) \leq d[v]$

מסקנה: יהיו $s, v \in V$ עבורם בריצת BellmanFord מתקיים $d[v] = \infty$ אזי לאחר כל רצף פעולות Relax נקבל כי $d[v] = \infty$

מסקנה: יהיו $s, v \in V$ עבורם בריצת BellmanFord מתקיים $d[v] = \delta(s, v)$ אזי לאחר כל רצף פעולות Relax נקבל כי $d[v] = \delta(s, v)$

למה: יהיו $s, t \in V$ עבורם בריצת BellmanFord מתקיים $d[s] = 0$ ויהי $\sigma \in \{s \rightarrow t\}$ מסלול אזי לאחר הפעלת הרצף Relax($\sigma[0], \sigma[1]$), ..., Relax($\sigma[n-1], \sigma[n]$) נקבל כי $d[t] \leq \ell(\sigma)$

למה: יהי $s \in V$ עבורו לא קיים מעגל שלילי אשר ניתן להגיע אליו מ- s אזי BellmanFord יוצא מהלולאה הראשית כאשר $i < |V|$ וכן מחזיר 0 וכן לכל $v \in V$ מתקיים $d[v] = \delta(s, v)$

למה: יהי $s \in V$ עבורו קיים מעגל שלילי אשר ניתן להגיע אליו מ- s אזי BellmanFord יוצא מהלולאה הראשית כאשר $i = |V|$ וכן מחזיר 1.

מסקנה: יהי $s \in V$ אזי

- (BellmanFord החזיר 1) \iff (קיים מעגל שלילי אשר ניתן להגיע אליו מ- s).
- (BellmanFord החזיר 0) \iff (לא קיים מעגל שלילי אשר ניתן להגיע אליו מ- s וכן לכל $v \in V$ מתקיים $d[v] = \delta(s, v)$).

למה: יהי $s \in V$ ויהי C מעגל בעץ BellmanFord באיזושהו שלב של הרצת BellmanFord אזי C מעגל שלילי.

למה: יהי $s \in V$ עבורו לא קיים מעגל שלילי אשר ניתן להגיע אליו מ- s אזי עץ BellmanFord הינו עץ G_π .

למה: יהי $s \in V$ עבורו קיים מעגל שלילי אשר ניתן להגיע אליו מ- s אזי עץ BellmanFord מכיל מעגל שלילי.

מסקנה: יהי $s \in V$ אזי BellmanFord פתרון לבעיית SSSP.

משפט: יהי $s \in V$ אזי אלגוריתם BellmanFord רץ בסיבוכיות $\mathcal{O}(|E| \cdot |V|)$.

הערה: נניח כי $\ell: E \rightarrow \mathbb{Z}$ וכן $\ell(e) \geq -W$ אזי קיים אלגוריתם לבעיית SSSP בסיבוכיות

$$\mathcal{O}(|E| \log^2(|V|) \log(|V| \cdot W) \log \log(|V|))$$

אלגוריתם לבדיקת קיום מעגל במשקל 0 בגרף מכוון חסר מעגלים שליליים: יהי G גרף מכוון חסר מעגלים שליליים אזי

```

function IsZeroCircle( $G, \ell$ ):
     $s \notin V$ 
     $V \leftarrow V \cup s$ 
    for  $v \in V \setminus \{s\}$  do
         $E \leftarrow E \cup \{(s, v)\}$ 
         $\ell((s, v)) \leftarrow 0$ 
    end
    BellmanFord( $G, \ell, s$ )
    for  $e \in E$  do
        if  $\delta(s, v) \neq \delta(s, u) + \ell(u, v)$  then
             $E \leftarrow E \setminus \{(s, v)\}$ 
        end
    end
    end
    if  $\exists \text{ circle } C \in G$  then
        return true
    end
    return false

```

טענה: בריצת IsZeroCircle לאחר מחיקת כל הקשתות נקבל את גרף מק"ב מ- s .

טענה: אם בריצת IsZeroCircle לאחר מחיקת כל הקשתות קיים מעגל C אזי $\ell(C) = 0$.

טענה: יהי C מעגל עבורו $\ell(C) = 0$ אזי בריצת IsZeroCircle לאחר מחיקת כל הקשתות נקבל כי C בגרף.

מסקנה: יהי G גרף מכוון חסר מעגלים שליליים אזי (G) בעל מעגל ממשקל 0 \iff IsZeroCircle מחזיר true.

טענה: יהי G גרף מכוון חסר מעגלים שליליים אזי סיבוכיות IsZeroCircle הינה $O(|V| \cdot |E|)$.

אלגוריתם למציאת מסלולים קצרים ביותר מנקודת מוצא בגרף מכוון אציקלי: יהי G מכוון אציקלי ויהי $s \in V$

```

function SSSP-DAG( $G, \ell, s$ ):
     $(d, \pi) \leftarrow \text{dict}(V)$ 
     $d[s] \leftarrow 0$ 
    for  $u \in V$  do
         $d[u] \leftarrow \infty$ 
         $\pi[u] \leftarrow \text{None}$ 
    end
    /* Knuth's Algorithm is an algorithm to compute a topological sorting. */
     $f \leftarrow \text{Knuth'sAlgorithm}(G)$ 
    for  $i \in [|V|]$  do
        for  $v \in \text{Adj}(f(i))$  do
            Relax( $(f(i), v)$ )
        end
    end
    end
    return  $(d, \pi)$ 

```

טענה: יהי G מכוון אציקלי ויהי $s \in V$ אזי SSSP-DAG(G) פתרון לבעיית SSSP.

טענה: יהי G מכוון אציקלי ויהי $s \in V$ אזי סיבוכיות SSSP-DAG(G) הינה $O(|E| + |V|)$.

אלגוריתם דייקסטרה למציאת מסלולים קצרים ביותר מנקודת מוצא בגרף ללא משקלים שליליים: יהי G גרף מכוון עבורו $\ell \geq 0$ ויהי

$s \in V$ אזי

```

function Dijkstra( $G, \ell, s$ ):
     $Q \leftarrow \text{heap}((V, \text{int}))$ 
     $(d, \pi) \leftarrow \text{dict}(V)$ 
     $d[s] \leftarrow 0$ 
    for  $u \in V$  do
         $d[u] \leftarrow \infty$ 
         $\pi[u] \leftarrow \text{None}$ 
    end
     $Q.\text{insert}((s, d[s]))$ 
    while  $Q \neq \emptyset$  do
         $u \leftarrow Q.\text{min}$ 
        for  $v \in \text{Adj}(u)$  do
            if  $d[v] = \infty$  then
                 $\pi[v] \leftarrow u$ 
                 $d[v] \leftarrow d[u] + \ell(u, v)$ 
                 $Q.\text{insert}((v, d[v]))$ 
            else if  $d[u] + \ell(u, v) < d[v]$  then
                 $\pi[v] \leftarrow u$ 
                 $d[v] \leftarrow d[u] + \ell(u, v)$ 
                 $Q.\text{decrease-key}((v, d[v]))$ 
            end
        end
    end
    return  $(d, \pi)$ 

```

למה: יהיו $s, u \in V$ עבורם בריצת Dijkstra הצומת u נמחקה מ- Q אזי $d[u] = \delta(s, u)$

משפט: יהי $s \in V$ אזי Dijkstra פתרון לבעיית SSSP כאשר $\ell \geq 0$

משפט: יהי $s \in V$ אזי ניתן לממש את Dijkstra עם Fibonacci heaps בסיבוכיות $\mathcal{O}(|E| + |V| \cdot \log(|V|))$

בעיית כל המסלולים הקצרים (APSP): יהי G גרף מכוון וממושקל ℓ אזי $D \in M_{|V|}(\mathbb{R})$ עבורו לכל $u, v \in V$ מתקיים $D_{u,v} = \delta(u, v)$ וכן $\Pi \in M_{|V|}(V)$ עבורו לכל $u, v \in V$ קיים מסלול קצר ביותר σ מ- u ל- v המקיים $(\Pi_{u,v}, v) \in \sigma$

פונקציית פוטנציאל: יהי G גרף אזי $p: V \rightarrow \mathbb{R}$

פונקציית משקל מותאמת: תהא p פונקציית פוטנציאל אזי פונקציית משקל ℓ_p עבורה לכל $u, v \in V$ המקיימים $(u, v) \in E$ מתקיים $\ell_p((u, v)) = \ell((u, v)) + p(u) - p(v)$

משפט: תהא p פונקציית פוטנציאל יהיו $s, t \in V$ ויהי σ מסלול מ- s ל- t אזי $\ell_p(\sigma) = \ell(\sigma) + p(s) - p(t)$

מסקנה: תהא p פונקציית פוטנציאל יהיו $s, t \in V$ ויהי σ מסלול מ- s ל- t אזי (σ) מסלול קצר ביותר ביחס ל- $\ell \iff$ (מסלול קצר ביותר ביחס ל- ℓ_p)

מסקנה: תהא p פונקציית פוטנציאל ויהי σ מעגל אזי $\ell_p(\sigma) = \ell(\sigma)$

מסקנה: תהא p פונקציית פוטנציאל ויהיו $s, t \in V$ אזי $\delta_\ell(s, t) = \delta_{\ell_p}(s, t) - p(s) + p(t)$

פונקציית פוטנציאל פיזבילית: יהי G גרף מכוון וממושקל ℓ אזי פונקציית פוטנציאל p עבורה $\ell_p \geq 0$

משפט: יהי G גרף מכוון וממושקל ℓ אזי (קיימת פונקציית פוטנציאל פיזבילית) $\iff (G)$ מצוייד עם ℓ חסר מעגלים שליליים.

אלגוריתם למציאת פונקציית פוטנציאל פיזבילית: יהי G גרף מכוון וממושקל ℓ אזי

```
function FeasiblePotential( $G, \ell$ ):
```

```

     $G' \leftarrow G \uplus \{s\}$ 
    for  $v \in V(G)$  do
         $E(G') \leftarrow E(G') \cup \{(s, v)\}$ 
         $\ell((s, v)) \leftarrow 0$ 
    end
     $c \leftarrow \text{BelmanFord}(G', \ell, s)$ 
    if  $c = 1$  then
        return None
    end
    else
         $p \leftarrow (V(G) \rightarrow \mathbb{R})$ 
        for  $v \in V(G)$  do
             $p(v) \leftarrow \delta(s, v)$ 
        end
        return  $p$ 
    end
end
```

טענה: יהי G גרף מכוון וממושקל ℓ אזי

• $(G \text{ מצוייד עם } \ell \text{ בעל מעגל שלילי}) \iff (\text{FeasiblePotential}(G, \ell) \text{ מחזיר None}).$

• $(G \text{ מצוייד עם } \ell \text{ בעל פונקציית פוטנציאל פיזבילית}) \iff (\text{FeasiblePotential}(G, \ell) \text{ מחזיר פונקציית פוטנציאל פיזבילית}).$

אלגוריתם ג'ונסון לבעיית כל המסלולים הקצרים: יהי G גרף מכוון וממושקל ℓ אזי

```
function Johnson( $G, \ell$ ):
```

```

     $p \leftarrow \text{FeasiblePotential}(G, \ell)$ 
    if  $p = \text{None}$  then
        return None
    end
     $\ell_p \leftarrow (E \rightarrow \mathbb{R})$ 
    for  $(u, v) \in E$  do
         $\ell_p((u, v)) = \ell((u, v)) + p(u) - p(v)$ 
    end
     $D_{\ell_p}, D_\ell \leftarrow M_{|V|}(\mathbb{R})$ 
     $\Pi \leftarrow M_{|V|}(E)$ 
    for  $v \in V$  do
         $(d, \pi) \leftarrow \text{Dijkstra}(G, \ell_p, v)$ 
        /* Here  $D$  and  $\Pi$  will be simplified, to get a solution to APSP as needed we can modify a bit Dijkstra's
           algorithm to calculate  $D$  and  $\Pi$  on the way to get constant time for this assignment. */
         $D_v \leftarrow d$ 
         $\Pi_v \leftarrow \pi$ 
    end
    for  $(u, v) \in E$  do
         $D_\ell((u, v)) = D_{\ell_p}((u, v)) - p(u) + p(v)$ 
    end
    return  $(D, \Pi)$ 
```

משפט: יהי G גרף מכוון וממושקל ℓ אזי $\text{Johnson}(G, \ell)$ פתרון לבעיית APSP.

טענה: יהי G גרף מכוון וממושקל ℓ אזי סיבוכיות הריצה של $\text{Johnson}(G, \ell)$ הינה $\mathcal{O}(|E||V| + |V|^2 \log(|V|))$.

מכפלת Min Plus: תהא $A \in M_{m \times n}(\mathbb{F})$ ותהא $B \in M_{n \times k}(\mathbb{F})$ אזי $A * B \in M_{m \times k}(\mathbb{F})$ באשר $(A * B)_{i,j} = \min_{k=1}^n (A_{i,k} + B_{k,j})$.

טענה: תהיינה $A, B \in M_n(\mathbb{F})$ אזי סיבוכיות הריצה של $A * B$ הינה $\mathcal{O}(n^3)$.

טענה: תהיינה $A, B, C \in M_n(\mathbb{F})$ אזי $(A * B) * C = (A * (B * C))$.

סימון: יהיו $s, v \in V$ אזי $\delta_k(s, v) = \min \{\ell(\sigma) \mid (\sigma \in \{s \rightarrow v\}) \wedge (|\sigma| \leq k)\}$.

טענה: יהיו $s, v \in V$ אזי $\delta_k(s, v) = \min_{u \in V} (\delta_{k-1}(s, u) + \ell(u, v))$.

סימון: יהי $s \in V$ אזי $\delta_k(s) \in M_{1 \times |V|}(\mathbb{R})$ באשר $(\delta_k(s))_v = \delta_k(s, v)$ לכל $v \in V$.

מטריצת המשקל: יהי G גרף מכוון וממושקל ℓ אזי $L \in M_{|V|}(\mathbb{R})$ באשר לכל $u, v \in V$ מתקיים

$$L_{u,v} = \begin{cases} 0 & u=v \\ \ell((u,v)) & (u \neq v) \wedge ((u,v) \in E) \\ \infty & (u \neq v) \wedge ((u,v) \notin E) \end{cases}$$

מסקנה: יהי $s \in V$ ותהא $L \in M_{|V|}(\mathbb{R})$ מטריצת המשקל אזי $\delta_k(s) = \delta_{k-1}(s) * L$

סימון: יהי $k \in \mathbb{N}$ אזי $D^{(k)} \in M_{|V|}(\mathbb{R})$ באשר לכל $u, v \in V$ מתקיים $D_{u,v}^{(k)} = \delta_k(u, v)$

מסקנה: תהא $L \in M_{|V|}(\mathbb{R})$ מטריצת המשקל אזי $D^{(k)} = D^{(k-1)} * L$

טענה: תהא $L \in M_{|V|}(\mathbb{R})$ מטריצת המשקל אזי $D^{(k)} = L^k$

הערה: $L^k = L * \dots * L$

טענה: יהי G גרף מכוון וממושקל ℓ וחסר מעגלים שליליים ויהיו $k, m \geq |V| - 1$ אזי $D^{(k)} = D^{(m)}$

טענה: יהי G גרף מכוון וממושקל ℓ בעל מעגל שלילי ויהי $v \in V$ המופיע במעגל שלילי אזי $D_{v,v}^{(|V|)} < 0$

מסקנה: תהא $L \in M_{|V|}(\mathbb{R})$ מטריצת המשקל אזי L^n פתרון לבעיית APSP.

אלגוריתם חזקה איטרטיבית: תהא $A \in M_n(\mathbb{R})$ תהא $*$ פעולה אסוציאטיבית אזי

function RepeatedSquaring($A, *$):

```

 $a_k \dots a_0 \leftarrow (n)_2$ 
 $B \leftarrow M_n(\mathbb{R})$ 
for  $i \in [k]$  do
    if  $a_i = 1$  then
         $B = B * A$ 
    end
     $A = A * A$ 
end
return  $B$ 
```

טענה: תהא $L \in M_{|V|}(\mathbb{R})$ מטריצת המשקל אזי $\text{RepeatedSquaring}(L, *)$ פתרון לבעיית APSP.

טענה: תהא $L \in M_{|V|}(\mathbb{R})$ מטריצת המשקל אזי סיבוכיות הריצה של $\text{RepeatedSquaring}(L, *)$ הינה $\mathcal{O}(|V|^3 \log(|V|))$

סימון: יהי $([n], E)$ גרף מכוון ויהי $k \in \mathbb{N}$ אזי $F^{(k)} \in M_n(\mathbb{R})$ באשר לכל $u, v \in V$ מתקיים

$F_{u,v}^{(k)} = \min \{ \ell(\sigma) \mid (\sigma \in \{u \rightarrow v\}) \wedge (\sigma \text{ עוברת דרך הצמתים } [k] \text{ למעט בהתחלה ובסוף}) \}$

סימון: יהי $([n], E)$ גרף מכוון ותהא $L \in M_n(\mathbb{R})$ מטריצת המשקל אזי $F^{(0)} \in M_n(\mathbb{R})$ באשר לכל $u, v \in V$ מתקיים $F_{u,v}^{(0)} = L$

טענה: יהי $([n], E)$ גרף מכוון ויהיו $u, v \in [n]$ אזי $F_{u,v}^{(k)} = \min \{ F_{u,v}^{(k-1)}, F_{u,k}^{(k-1)} + F_{k,v}^{(k-1)} \}$

אלגוריתם פלוגיד-וורשאל: יהי $([n], E)$ גרף מכוון ותהא $L \in M_n(\mathbb{R})$ מטריצת המשקל אזי

```

function FloydWarshall( $n, L$ ):
     $\Pi \leftarrow M_n([n])$ 
    for  $u \in [n]$  do
        for  $v \in [n]$  do
            if  $(u \neq v) \wedge (L_{u,v} < \infty)$  then
                 $\Pi_{u,v} \leftarrow u$ 
            else
                 $\Pi_{u,v} \leftarrow \text{None}$ 
            end
        end
    end
     $F \leftarrow L$ 
    for  $k \in [n]$  do
        for  $u \in [n]$  do
            for  $v \in [n]$  do
                if  $F_{u,k} + F_{k,v} < F_{u,v}$  then
                     $F_{u,v} \leftarrow F_{u,k} + F_{k,v}$ 
                     $\Pi_{u,v} \leftarrow \Pi_{k,v}$ 
                end
            end
        end
    end
    return  $(F, \Pi)$ 

```

טענה: יהי $([n], E)$ גרף מכוון ותהא $L \in M_n(\mathbb{R})$ מטריצת המשקל אזי FloydWarshall(n, L) פתרון לבעיית APSP.

טענה: יהי $([n], E)$ גרף מכוון ותהא $L \in M_n(\mathbb{R})$ מטריצת המשקל אזי סיבוכיות הריצה של FloydWarshall(n, L) הינה $\mathcal{O}(n^3)$.

קבוצה בלתי תלויה: יהי G גרף אזי $I \subseteq V$ עבורה לכל $u, v \in I$ מתקיים $(u, v) \notin E$.

סימון: יהי $([n], E)$ גרף שרוך ויהי $w : [n] \rightarrow \mathbb{R}_{\geq 0}$ אזי $\text{mis}(i) = \max \{w(I) \mid (I \subseteq [i]) \wedge (I \text{ בלתי תלויה})\}$

טענה: יהי $([n], E)$ גרף שרוך ויהי $w : [n] \rightarrow \mathbb{R}_{\geq 0}$ אזי $\text{mis}(0) = 0$ וכן $\text{mis}(1) = w(1)$ וכן

$$\text{mis}(i) = \max \{w(i) + \text{mis}(i-2), \text{mis}(i-1)\}$$

מסקנה: יהי $([n], E)$ גרף שרוך ויהי $w : [n] \rightarrow \mathbb{R}_{\geq 0}$ אזי $\text{mis}(n)$ בעל סיבוכיות זמן $\mathcal{O}(n)$.

תת־סדרה: יהי Σ אלפבית ותהא $A \in \Sigma^*$ אזי $B \in \Sigma^*$ עבורה קיימת $f : [|B|] \rightarrow [|A|]$ עולה ממש וחס"ע המקיימת $A_{f(i)} = B_i$ לכל $i \in [|B|]$.

סימון: יהי Σ אלפבית תהא $A \in \Sigma^*$ ותהא $B \in \Sigma^*$ תת־סדרה אזי $B \triangleleft A$.

בעיית תת־סדרה משותפת ארוכה ביותר (LCS): יהי Σ אלפבית ותהינה $A, B \in \Sigma^*$ אזי $\max \{|C| \mid (C \in \Sigma^*) \wedge (C \triangleleft A) \wedge (C \triangleleft B)\}$.

סימון: תהינה $A, B \in \Sigma^*$ תהא $k \leq |A|$ ותהא $\ell \leq |B|$ אזי $\text{lcs}(k, \ell) = \max \{|C| \mid (C \triangleleft (A_1, \dots, A_k)) \wedge (C \triangleleft (B_1, \dots, B_\ell))\}$.

$$\text{lcs}(k, \ell) = \begin{cases} 0 & (k=0) \vee (\ell=0) \\ \text{lcs}(k-1, \ell-1)+1 & (k, \ell > 0) \wedge (A_k = B_\ell) \\ \max\{\text{lcs}(k-1, \ell), \text{lcs}(k, \ell-1)\} & (k, \ell > 0) \wedge (A_k \neq B_\ell) \end{cases}$$

מסקנה: תהינה $A, B \in \Sigma^*$ אזי $\text{lcs}(|A|, |B|)$ בעל סיבוכיות זמן $\mathcal{O}(|A| \cdot |B|)$ וסיבוכיות מקום $\mathcal{O}(|A| \cdot |B|)$.

בעיית תת־סדרה עולה ארוכה ביותר (LIS): יהי Σ אלפבית בעל סדר \prec ותהא $A \in \Sigma^*$ אזי $\max \{|C| \mid (C \triangleleft A) \wedge (\forall i. C_{i-1} \prec C_i)\}$.

טענה: תהא $A \in \Sigma^*$ אזי בעיית LIS של A הינה בעיית LCS של $(A, \text{sort}(A))$.

סימון: תהא $A \in \Sigma^*$ אזי $\text{lenlis}(k) = \max \{|X| \mid ((A_1, \dots, A_k) \text{ הינו lis של } X) \wedge (A_k \text{ מסתיים עם } X)\}$.

טענה: תהא $A \in \Sigma^*$ אזי $\text{lenlis}(1) = 1$ וכן $\text{lenlis}(k) = \max_{i \in [k-1]} \{\text{lenlis}(i) \mid A_i \prec A_k\}$.

סימון: תהא $A \in \Sigma^*$ אזי $\pi_{\text{lis}}(1) = \text{None}$ וכן $\pi_{\text{lis}}(k) = \arg \max \{\text{lenlis}(i) \mid A_i \prec A_k\}$.

מסקנה: תהא $A \in \Sigma^*$ ויהי $k = \arg \max \{\text{lenlis}(1), \dots, \text{lenlis}(|A|)\}$ אזי $(x_{\pi_{\text{lis}}(\ell)(k)}, \dots, x_{\pi_{\text{lis}}(2)(k)}, x_{\pi_{\text{lis}}(1)(k)}, x_k)$ פתרון של LIS

בעל סיבוכיות $\mathcal{O}(|A|^2)$.

סימון: תהא $A \in \Sigma^*$ אזי $\min \text{lis}(m) = \min \{x_k \mid \text{lenlis}(k) = m\}$.

טענה: תהא $A \in \Sigma^*$ אזי $\min \text{lis}$ עולה ממש.

מסקנה: תהא $A \in \Sigma^*$ אזי $(\min \text{lis}(1), \dots, \min \text{lis}(\ell))$ פתרון של LIS בעל סיבוכיות ריצה $\mathcal{O}(|A| \cdot \log(|A|))$.

סימון: יהיו $p_1 \dots p_n \in (0, 1]$ ויהי T עץ חיפוש בינארי מעל $\{x_1 \dots x_n\}$ אזי $\text{costp}(T) = \sum_{i=1}^n (p_i \cdot \text{depth}_T(x_i))$.

בעיית עץ חיפוש בינארי סטטי אופטימלי: יהיו $p_1 \dots p_n \in (0, 1]$ אזי עץ חיפוש בינארי T עבורו $\text{costp}(T)$ מינימלי.

טענה: יהיו $p_1 \dots p_n \in (0, 1]$ ויהי T עץ חיפוש בינארי אזי $\text{costp}(T) = (\sum_{i=1}^n p_i) + \text{costp}(T.\text{left}) + \text{costp}(T.\text{right})$
מסקנה: יהיו $p_1 \dots p_n \in (0, 1]$ ויהי T פתרון לבעיית עץ חיפוש בינארי סטטי אופטימלי אזי $T.\text{left}, T.\text{right}$ הינם פתרונות לבעיית עץ חיפוש בינארי סטטי אופטימלי.

סימון: יהיו $p_1 \dots p_n \in (0, 1]$ אזי $\text{pp}(i, j) = \sum_{k=i}^j p_k$

סימון: יהיו $p_1 \dots p_n \in (0, 1]$ ויהיו $x_1 \dots x_n$ אזי T עץ חיפוש בינארי מעל $\{x_i \dots x_j\}$ אזי $\text{cp}(i, j) = \min \{\text{costp}(T) \mid \{x_i \dots x_j\}\}$

טענה: יהיו $p_1 \dots p_n \in (0, 1]$ ויהיו $x_1 \dots x_n$ אזי $\text{cp}(i, i) = p_i$ וכן $\text{cp}(i, i-1) = 0$

$\text{cp}(i, j) = \text{pp}(i, j) + \min_{i \leq k \leq j} (\text{cp}(i, k-1) + \text{cp}(k+1, j))$

מסקנה אלגוריתם לבעיית עץ חיפוש בינארי סטטי אופטימלי: יהיו $p_1 \dots p_n \in (0, 1]$ ויהיו $x_1 \dots x_n$ אזי

```
function OSBST(pp):
  C ← List([n]2)
  for i ← [n+1] do
    | C(i, i-1) ← 0
  end
  for d ← {0, ..., n-1} do
    for i ← [n-d] do
      C(i, i+d) ← ∞
      for k ← {i, ..., i+d} do
        t ← pp(i, j) + C(i, k-1) + C(k+1, j)
        if t < C(i, j) then
          C(i, j) ← t
          K(i, j) ← k
        end
      end
    end
  end
end
```

מסקנה: יהיו $p_1 \dots p_n \in (0, 1]$ אזי K OSBST(pp) משרה פתרון לבעיית עץ חיפוש בינארי סטטי אופטימלי.

מסקנה: יהיו $p_1 \dots p_n \in (0, 1]$ אזי OSBST(pp) בעל סיבוכיות ריצה $\mathcal{O}(n^3)$.

הערה: קיים אלגוריתם קנות' לבעיית עץ חיפוש בינארי סטטי אופטימלי בסיבוכיות $\mathcal{O}(n^2)$.

בעיית 0/1 תרמיל הגב: יהיו $W, w_1 \dots w_n, v_1 \dots v_n \geq 0$ באשר $S \subseteq [n]$ $\sum_{i \in S} v_i \leq W$ מקסימלית וכן $\sum_{i \in S} w_i \leq W$.

בעיית שבר תרמיל הגב: יהיו $W, w_1 \dots w_n, v_1 \dots v_n \geq 0$ אזי $f: [n] \rightarrow [0, 1]$ באשר $\sum_{i \in [n]} f(i) v_i \leq W$ וכן $\sum_{i \in [n]} f(i) w_i \leq W$.

אלגוריתם חסמן לבעיית שבר תרמיל הגב: יהיו $W, w_1 \dots w_n > 0$ ויהיו $v_1 \dots v_n \geq 0$ אזי

סימון: יהיו $W, w_1 \dots w_n, v_1 \dots v_n \geq 0$ אזי $\text{bknap}(k, W) = \max \{ \sum_{i \in S} v_i \mid (S \subseteq [k]) \wedge (\sum_{i \in S} w_i \leq W) \}$

טענה: יהיו $w_1 \dots w_n, v_1 \dots v_n \geq 0$ אזי

• יהי $m \geq 0$ אזי $\text{bknap}(0, m) = 0$

• יהי $i \in [n]$ אזי $\text{bknap}(i, 0) = 0$

• יהי $m \geq 0$ ויהי $i \in [n]$ אזי $\text{bknap}(i, m) = \begin{cases} \text{bknap}(i-1, m) & w_i > m \\ \max \{ \text{bknap}(i-1, m), \text{bknap}(i-1, m-w_i) + v_i \} & w_i \leq m \end{cases}$

מסקנה: יהיו $W, w_1 \dots w_n, v_1 \dots v_n \geq 0$ אזי חישוב $\text{bknap}(n, W)$ בעל סיבוכיות ריצה $\mathcal{O}(nW)$.

מסקנה אלגוריתם לבעיית 0/1 תרמיל הגב: יהיו $W, w_1 \dots w_n, v_1 \dots v_n \geq 0$ אזי

מסקנה: יהיו $W, w_1 \dots w_n, v_1 \dots v_n \geq 0$ אזי $\text{ZeroOneKnapsack}(W, w_1 \dots w_n, v_1 \dots v_n)$ פתרון לבעיית 0/1 תרמיל הגב.


```

function FractionalKnapsack( $W, w_1, \dots, w_n, v_1, \dots, v_n$ ):
     $f \leftarrow ([n] \rightarrow [0, 1])$ 
     $P \leftarrow \text{List}([n] \times \mathbb{R})$ 
    for  $i \leftarrow [n]$  do
         $P(i) \leftarrow (i, \frac{v_i}{w_i})$ 
         $f(i) \leftarrow 0$ 
    end
     $P \leftarrow \text{sort}(P)$  // Sort from high to low based on second coordinate.
     $t \leftarrow 0$ 
     $i \leftarrow 1$ 
    while  $(t < W) \wedge (i \leq n)$  do
         $j \leftarrow P(i)[0]$ 
        if  $t + w_j \leq W$  then
             $f(j) \leftarrow 1$ 
             $t \leftarrow t + w_j$ 
        end
        else
             $f(j) \leftarrow \frac{W-t}{w_j}$ 
             $t \leftarrow W$ 
        end
    end
    return  $f$ 

```

```

function ZeroOneKnapsack( $W, w_1, \dots, w_n, v_1, \dots, v_n$ ):
     $k \leftarrow n$ 
     $w \leftarrow W$ 
     $S \leftarrow \text{Set}([n])$ 
     $S \leftarrow \emptyset$ 
    while  $(k > 0) \wedge (w > 0)$  do
        if  $bknap(k, w) \neq bknap(k-1, w)$  then
             $S \leftarrow S \cup \{k\}$ 
             $k \leftarrow k - 1$ 
             $w \leftarrow w - w_k$ 
        else
             $k \leftarrow k - 1$ 
        end
    end

```

רשת זרימה: יהי G גרף מכוון וממושקל $c \geq 0$ ותהייה $s, t \in V$ אזי (V, E, c, s, t) .

פונקציית קיבולת: תהא (V, E, c, s, t) רשת זרימה אזי c .

קודקוד מקור: תהא (V, E, c, s, t) רשת זרימה אזי s .

קודקוד בור: תהא (V, E, c, s, t) רשת זרימה אזי t .

פונקציית זרימה: תהא (V, E, c, s, t) רשת זרימה אזי $f : E \rightarrow \mathbb{R}_{\geq 0}$ עבורה

• חסם קיבולת: $f \leq c$.

• שימור זרם: לכל $v \in V \setminus \{s, t\}$ מתקיים $\sum_{(u,v) \in E} f((u,v)) = \sum_{(v,u) \in E} f((v,u))$.

בעיית הזרימה המקסימלית: תהא (V, E, c, s, t) רשת זרימה אזי פונקציית זרימה f עבורה $\sum_{(u,t) \in E} f((u,t)) - \sum_{(t,u) \in E} f((t,u))$ מקסימלית.

חתך s-t: תהא (V, E, c, s, t) רשת זרימה אזי (S, T) באשר $S, T \subseteq V$ וכן $S \cup T = V$ וכן $s \in S$ וכן $t \in T$.

קשתות חוצות: תהא G רשת זרימה ויהי (S, T) חתך s-t אזי $E(S, T) = \{(u, v) \in E \mid (u \in S) \wedge (v \in T)\}$.

קשתות אחוריות: תהא G רשת זרימה ויהי (S, T) חתך s-t אזי $E(T, S) = \{(u, v) \in E \mid (u \in T) \wedge (v \in S)\}$.

קיבולת של חתך: יהי (S, T) חתך s-t אזי $c(S, T) = \sum_{e \in E(S, T)} c(e)$.

זרימה על פני חתך: יהי (S, T) חתך s-t אזי $f(S, T) = \sum_{e \in E(S, T)} f(e) - \sum_{e \in E(T, S)} f(e)$.

ערך/גודל של זרימה: תהא f זרימה אזי $|f| = f(V \setminus \{t\}, \{t\})$.

למה: תהא f זרימה ויהי (S, T) חתך s-t אזי $|f| = f(S, T)$.

מסקנה: תהא f זרימה אזי $|f| = f(\{s\}, V \setminus \{s\})$.

למה: תהא f זרימה ויהי (S, T) חתך s-t אזי $f(S, T) \leq c(S, T)$.

חתך s-t מינימלי: תהא f זרימה אזי (S, T) חתך s-t עבורו $f(S, T) = c(S, T)$.

מסקנה: תהא f זרימה ויהי (S, T) חתך s-t מינימלי אזי f זרימה מקסימלית.

מסלול ניתן להגדלה s-t: תהא f זרימה אזי $P \in \{s \rightarrow t\}$ באשר $f(e) < c(e)$ לכל $e \in P$.

טענה הגדלת מסלול: תהא f זרימה ויהי $P \in \{s \rightarrow t\}$ מסלול ניתן להגדלה s-t אזי קיימת פונקציית זרימה g עבורה $g|_{E \setminus P} = f|_{E \setminus P}$ וכן $|g| < |f|$.

זרימה חוסמת: פונקציית זרימה f עבורה לא קיים מסלול ניתן להגדלה s-t.

קשת אנטי-מקבילה: יהי G גרף מכוון ותהא $e \in E$ עבורה $e^{-1} \in E$ אזי e^{-1} .

רשת זרימה השוורית: תהא (V, E, c, s, t) רשת זרימה חסרת קשתות אנטי-מקבילות ותהא f זרימה אזי (V, E_f, c_f, s, t) באשר

• $E_f = \{e \in E \mid c(e) > f(e)\} \cup E^{-1}$.

• פונקציית שווריות הקיבולת: תהא $e \in E_f$ אזי $c_f(e) = \begin{cases} c(e) - f(e) & e \in E \\ f(e^{-1}) & e \in E^{-1} \end{cases}$.

מסלול ניתן לשיפור s-t: תהא f זרימה אזי מסלול $P \in \{s \rightarrow t\}$ בגרף G_f .

מחסום/שווריות הקיבולת של מסלול: תהא f זרימה ויהי P מסלול ניתן לשיפור s-t אזי $c_f(P) = \min \{c_f(e) \mid e \in P\}$.

זרימה משופרת: תהא f זרימה ויהי P מסלול ניתן לשיפור s-t אזי $f_P(e) = \begin{cases} f(e) + c_f(P) & e \in P \\ f(e) - c_f(P) & e^{-1} \in P \\ f(e) & \text{else} \end{cases}$ לכל $e \in E(G)$.

למה: תהא f זרימה ויהי P מסלול ניתן לשיפור s-t אזי f_P זרימה של G וכן $|f_P| = |f| + c_f(P)$.

משפט: תהא f זרימה התב"ש

• f זרימה מקסימלית ב- G .

• לכל מסלול $P \in \{s \rightarrow t\}$ בגרף G_f מתקיים כי P אינו מסלול ניתן לשיפור s-t.

• קיים (S, T) חתך s-t מינימלי ל- G .

אלגוריתם פורד-פלקרסון: תהא (V, E, c, s, t) רשת זרימה אזי

סימון: תהא (V, E, c, s, t) רשת זרימה אזי $\text{FF} = \text{FordFulkerson}(V, E, c, s, t)$.

משפט: תהא (V, E, c, s, t) רשת זרימה באשר $c(E) \subseteq \mathbb{N}$ אזי קיימת זרימה מקסימלית f באשר $f(E) \subseteq \mathbb{N}$.

טענה: תהא (V, E, c, s, t) רשת זרימה באשר $c(E) \subseteq \mathbb{N}$ אזי בכל איטרציה של FF מתקיים

• f זרימה של G .

• $f(E) \subseteq \mathbb{N}$.

• $c_f(P) \geq 1$.

משפט: תהא (V, E, c, s, t) רשת זרימה באשר $c(E) \subseteq \mathbb{N}$ ותהא f זרימה מקסימלית באשר $f(E) \subseteq \mathbb{N}$ אזי

```

function FordFulkerson( $V, E, c, s, t$ ):
     $f \leftarrow (E \rightarrow \mathbb{R})$ 
     $f \leftarrow 0$ 
    while True do
         $G_f \leftarrow \text{ResidualNetwork}(G, c, s, t, f)$  // Construct it like any graph.
         $\pi_{G_f} \leftarrow \text{BFS}(G, s)$ 
        if  $\{s \rightarrow t\} \cap \pi_{G_f} = \emptyset$  then
            | return  $f$ 
        else
            |  $P \leftarrow \{s \rightarrow t\} \cap \pi_{G_f}$  // The path is taken from  $\pi_{G_f}$ .
            |  $f \leftarrow f_P$ 
    end

```

• FF פתרון לבעיית הזרימה המקסימלית לכל בחירת מסלולים באלגוריתם.

• FF עושה לכל היותר $|f|$ שיפורי מסלול.

• $\text{FF}(E) \subseteq \mathbb{N}$.

מסקנה: תהא (V, E, c, s, t) רשת זרימה באשר $c(E) \subseteq \mathbb{N}$ ותהא f זרימה מקסימלית באשר $f(E) \subseteq \mathbb{N}$ אזי סיבוכיות זמן הריצה של FF הינה $\mathcal{O}(|E| |f|)$.

משפט זרימה מקסימלית קיבולת מינימלית: תהא G רשת זרימה אזי $\{c(S, T) \mid \text{חתך } s\text{-}t \text{ מינימלי}\} = \max\{|f| \mid f \text{ זרימה}\}$.