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, π, color) ← dict(V(G))
    for u ∈ V(G)\{s} do
        color[u] ← White
        d[u] ← ∞
        π[u] ← Null
    end
    color[s] ← Grey
    d[s] ← 0
    π[s] ← Null
    Q ← queue()
    while Q ≠ ∅ do
        u ← Q.head
        for v ∈ Neighbor(u) do
            if color[v] = White then
                color[v] ← Grey
                d[v] ← d[u] + 1
                π[v] ← u
                Q.enqueue(v)
            end
        end
        Q.dequeue()
        color[u] ← Black
    end
    return (d, π, color)
```

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

משפט: יהי G גרף ויהי $s\in V$ אזי $\{v\in V\mid \text{BFS}(G,s).\text{color}[v]=\text{Black}\}=[s]_\to$.

סימון: יהי G גרף ויהיו $u,v\in V$ אזי $\delta(v,u)=\min(\{\text{len}(\sigma)\mid v,u \text{ טיול בין } \sigma\})$.

טענה: יהי G גרף ויהיו $v,u,w\in V$ באשר $(w,u)\in E$ אזי $\delta(v,u)\le\delta(v,w)+1$.

למה: יהי G גרף ויהיו $s,v\in V$ אזי בכל שלב בהרצת $\text{BFS}(G,s)$ מתקיים $d[v]\ge\delta(v)$.

למה: יהי G גרף יהי שלב בהרצת $\text{BFS}(G,s)$ בו $Q=(v_1\dots v_n)$ אזי מתקיים $d[v_i]\le d[v_{i+1}]$ וכן $d[v_i]\le d[v_1]+1$.

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 $V_\pi=\{v\in V\mid \text{BFS}(G,s).\pi[v]\ne\text{Null}\}\cup\{s\}$ וכן $E_\pi=\{(\pi[v],v)\mid v\in V_\pi\backslash\{s\}\}$ 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)$\Longleftrightarrow$(לכל $v\in V$ מתקיים $\deg(v)\in\mathbb{N}_{\text{even}}$).

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):
    σ ← List(E(G))
    u ← Neighbor(v)
    while u ≠ v do
        σ.append({v, u})
        G = G\{{v, u}}
        u ← Neighbor(u)
    end
    if length(σ) = |E(G)| then
        return σ
    end
    else
        w ← {x ∈ V(G) | (∃y ∈ V(G).(x, y) ∈ σ) ∧ (deg(x) > 0)}
        σ[w] = EulerCircle(G, w)
    end
    return σ
```

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 EulerCircle (G, v) הינה $\mathcal{O}(|E|)$.

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

משפט: יהי G גרף קשיר ולא מכוון עבורו לכל $v \in V$ מתקיים $\deg(v) \in \mathbb{N}_{\text{even}}$ אזי 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} ← {v ∈ V(G) | deg(v) ∈ N_odd}
    G = G + {{v, u}}
    σ = EulerCircle(G, v)
    return σ\{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, π, color) ← BFS(G)
    for (v, u) ∈ 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)$\iff$(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 $|\sigma| = \min\{|\tau| \mid t\text{־ל } s\text{־מסלול מ } \tau\}$.

גרף המסלולים הקצרים ביותר מקודקוד (גרף מק"ב): יהי G גרף ויהי $s \in V$ נגדיר $E' = \{e \in E \mid s\text{־חלק ממסלול קצר ביותר היוצא מ } e\}$ אזי (V, E').

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

```
function ShortestPathGraph(G, s):
    (d, π, color) ← BFS(G)
    E' ← E(G_π)
    for (u, v) ∈ E(G) do
        if |height_{G_π}(u) − height_{G_π(v)}| = 1 then
            E'.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 $\text{ShortestPathGraph}\left(G, s\right)$ הינו גרף מק"ב מ־s.

גרף המסלולים הקצרים ביותר בין קודקודים: יהי G גרף ויהיו $s, t \in V$ נגדיר $E' = \{e \in E \mid t\text{ חלק ממסלול קצר ביותר היוצא מ־}s\text{ ל־} e\}$ 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 $\mathcal{O}\left(|V| + |E|\right)$.

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

```
function DFS(G, s):
    (k, π) ← dict(V)
    color ← dict(E)
    k[s] ← 1
    π[s] ← Null
    for u ∈ V\s do
        k[u] ← 0
        π[u] ← Null
    end
    for e ∈ E do
        color[e] ← White
    end
    i ← 2
    v ← s
    while (∃u ∈ Adj(v).color[(v, u)] = White) ∨ (π[v] ≠ Null) do
        if {u ∈ Adj(v) | color[(v, u)] = White} ≠ ∅ then
            w ← {u ∈ Adj(v) | color[(v, u)] = White}
            color[(v, w)] ← Black
            if k[w] = 0 then
                k[w] ← i
                π[w] ← v
                v ← w
                i ← i + 1
            end
        else
            v ← π[v]
        end
    end
    return (k, π)
```

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

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

טענה: יהי G גרף ויהיו $s, v \in V$ באשר $v \in [s]_{\to}$ אזי בהרצת $\text{DFS}\left(G, s\right)$ מתקיים $k\left[v\right] > 0$.

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

טענה: עץ DFS הינו עץ.

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

- קשתות עץ: קשת $e \in E(G)$ עבורה $e \in E(G_\pi)$.
- 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
    color[u] ← Black
    i ← i + 1
    f[v] ← i
```

זמן נסיגה: יהי G גרף ויהי $s \in V(G)$ אזי f בהרצת DFS(G).

טענה Gray Path Lemma: יהיו $u,v \in V$ אזי (v צאצא של u ביער G_π)$\Longleftrightarrow$($k[u] < k[v] < f[u]$).

טענה: יהיו $u,v \in V$ אזי ((u,v) קשת חוצה)$\Longleftrightarrow$($f[v] < k[u]$).

משפט הסוגריים: יהי G גרף ויהיו $u,v \in V$ אזי מתקיים בדיוק אחד מהבאים

- מתקיים $[k(u), f(u)] \cap [k(v), f(v)] = \varnothing$ 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 (v צאצא של u ביער G_π)$\Longleftrightarrow$(בזמן $k(u)$ באלגוריתם 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)$\Longleftrightarrow$(קיים מיון טופולוגי על G).

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

משפט: יהי G גרף מכוון אזי (G רציקלי)$\Longleftrightarrow$(אין קשתות אחוריות ב־G).

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

קודקוד מנתק: יהי G גרף מכוון אזי $v\in V\left(G\right)$ עבורו $\left|G/\underset{G}{\longrightarrow}\right| \le \left|G-\{v\}/\underset{G-\{v\}}{\longrightarrow}\right|$.

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 $\text{DFS}\left(G\right)$.

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

function `DetachableVertices`(G):
- $s \leftarrow V$
- $(k, f, \pi, \text{low}) \leftarrow \text{DFS}(G, s)$
- $A \leftarrow \text{set}(V)$
- **if** $|Adj_{G_\pi}(s)| \neq 1$ **then**
  - $A.\text{append}(s)$
- **end**
- **for** $u \in V\backslash\{s\}$ **do**
  - **if** $\exists v \in children(u).low[v] \geq k[u]$ **then**
    - $A.\text{append}(u)$
  - **end**
- **end**
- **return** A

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

טענה: יהי G גרף מכוון וקשיר אזי $\text{DetachableVertices}\left(G\right)$ 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 רק"ה של G)$\Longleftrightarrow$(C רק"ה של G^T).

אלגוריתם למציאת רכיבים קשירים היטב: יהי G גרף מכוון אזי

function `SCC`(G):
- $(k, f, \pi) \leftarrow \text{DFS}(G)$
- `/* In the next line, in the main loop of DFS we go through the vertices in decreasing order of f[u] */`
- $(k', f', \pi') \leftarrow \text{DFS}(G^T)$
- $A \leftarrow \text{set}(\text{set}(V))$
- **for** $v \in V$ **do**
  - $A.\text{append}\left([v]_{\underset{G_\pi^T}{\longrightarrow}}\right)$
- **end**
- **return** A

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

אלגוריתם למציאת גרף הרכיבים: יהי G גרף מכוון אזי

function ComponentGraph(G):

- $V^* \leftarrow \text{SCC}(G)$
- $E^* \leftarrow \text{set}((V^*)^2)$
- **for** $(u,v) \in E$ **do**
  - **if** $[v]_{\underset{G_\pi^T}{\longrightarrow}} \neq [u]_{\underset{G_\pi^T}{\longrightarrow}}$ **then**
    - $E^*.\text{append}\left(\left([v]_{\underset{G_\pi^T}{\longrightarrow}}, [u]_{\underset{G_\pi^T}{\longrightarrow}}\right)\right)$
  - **end**
- **end**
- **return** (V^*, E^*)

למה: יהי G גרף מכוון אזי G^* אציקלי.

הגדרה: יהי G גרף ותהא $U \subseteq V$ אזי

- 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 ויהיו $C_1, C_2 \subseteq V$ רק"ה באשר $(C_1, C_2) \in E(G^*)$ אזי $f(C_2) < f(C_1)$.

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

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

קבוצת מוצא: יהי G גרף מכוון אזי $S \subseteq V$ המקיימת $\forall v \in V.\exists s \in S.s \to v$.

אלגוריתם למציאת קבוצת מוצא מינימלית: יהי G גרף מכוון אזי

function MinimalOriginSet(G):

- $A \leftarrow \text{set}(V(G))$
- $G^* \leftarrow \text{ComponentGraph}(G)$
- **for** $C \in V(G^*)$ **do**
  - $v \leftarrow \{u \in C \mid \nexists w \in V(G)\backslash C.(w,u) \in E(G)\}$
  - $A.\text{append}(v)$
- **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 $w : E \to \mathbb{R}$ 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 $T \leq G$ עבורו $w(T) = \min\{w(S) \mid G\text{של עץ פורש } S\}$.

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

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

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

טענה: יהי $T \leq G$ עץ פורש תהא $e_1 \in E(G) \backslash E(T)$ 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 $V(G) \setminus [v]_{\underset{T-\{e\}}{\longrightarrow}}, [v]_{\underset{T-\{e\}}{\longrightarrow}}$ חתך של G.

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

```
function MST(G, w):
    color ← dict(E)
    for e ∈ E do
        color[e] = White
    end
    while ∃e ∈ E.color[e] = White do
        Blueless ← {A ⊆ V | ∀e ∈ (A² ∩ E).color[e] ≠ Blue}
        Redless ← {σ circle in G | ∀i ∈ [len(σ)].color[(σ[i], σ[i+1])] ≠ Red}
        if Blueless ≠ ∅ then
            A ← Blueless
            f ← argmin_{e∈A²∩E}(w(e))
            color[f] = Blue
        end
        if Redless ≠ ∅ then
            σ ← Redless
            f ← argmax_{e∈σ}(w(e))
            color[f] = Red
        end
    end
    return (V, {e ∈ E | color[e] = Blue})
```

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

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

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

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

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

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

```
function Prim'sAlgorithm(G):
    color ← dict(E)
    U ← set(V)
    for e ∈ E do
        color[e] = White
    end
    r ← V
    U.append(r)
    while U ≠ V do
        (u, v) ← argmin_{e∈U×(V\U)}(w(e))
        color[(u, v)] = Blue
        U.append(v)
        for w ∈ U do
            if (w, v) ∈ E then
                color[(w, v)] = Red
            end
        end
    end
    return (V, {e ∈ E | color[e] = Blue})
```

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

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

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

הערה: יהי G גרף קשיר לא מכוון וממושקל w אזי ניתן לממש את $\text{Prim'sAlgorithm}\left(G\right)$ בסיבוכיות $\mathcal{O}\left(\left|E\right|+\left|V\right|\log\left|V\right|\right)$.

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

```
function Kruskal'sAlgorithm(G):
    color ← dict(E)
    L ← sort(E)
    for (u,v) ∈ L do
        if ∃σ ∈ {u → v}.∀i ∈ [n].color(σ(i)) = Blue then
            color[e] = Red
        end
        else
            color[e] = Blue
        end
    end
    return (V, {e ∈ E | color[e] = Blue})
```

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

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

משפט: יהי G גרף קשיר לא מכוון וממושקל w אזי ניתן לממש את $\text{Kruskal'sAlgorithm}\left(G\right)$ עם Union-Find בסיבוכיות $\mathcal{O}\left(\left|E\right|\log\left|V\right|\right)$ וכן סיבוכיות amortized $\mathcal{O}\left(\left|E\right|\cdot\alpha\left(\left|V\right|\right)\right)$.

אלגוריתם Borůvska למציאת עץ פורש מינימלי: יהי G גרף קשיר לא מכוון וממושקל w באשר w חח"ע אזי

```
function Borůvska'sAlgorithm(G):
    Trees ← set(set(G))
    for v ∈ V do
        Trees.append({v})
    end
    while |Trees| ≠ 1 do
        for T ∈ Tree do
            (u,v) ← argmin_{(u,v)∈V(T)×V(G)}(w((u,v)))
            S ← {S ∈ Tree | u ∈ V(S)}
            S ← S + T + {(u,v)}
            Trees.Remove(T)
        end
    end
    A ← Trees
    return A
```

טענה: יהי G גרף קשיר לא מכוון וממושקל w באשר w חח"ע אזי סיבוכיות $\text{Borůvska'sAlgorithm}\left(G\right)$ הינה $\mathcal{O}\left(\left|E\right|\log\left|V\right|\right)$.

משפט: יהי G גרף קשיר לא מכוון וממושקל w באשר w חח"ע אזי קיים ויחיד $T\le G$ עפ"מ.

משפט: יהי G גרף קשיר לא מכוון וממושקל w באשר w חח"ע אזי $\text{Borůvska'sAlgorithm}\left(G\right)$ עפ"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 עפ"מ $T\le G$ עבורו $A\subseteq E\left(T\right)$ וכן $e\notin E\left(T\right)$.

טענה: יהיו $T_1,T_2\le G$ עפ"מ ויהיו $\alpha_1\le\ldots\le\alpha_n$ ו־$\beta_1\le\ldots\le\beta_m$ 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):
    ε ← min({w(e_1) − w(e_2) | (e_1, e_2 ∈ E) ∧ (w(e_1) ≠ w(e_2))})/2
    for e ∈ E do
        if e ∈ F then

        end
        else
            w'(e) ← w(e)
        end
    end
    return Kruskal'sAlgorithm(G, w')
```

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

מסקנה: תהא $E \subseteq F$ אזי $\text{PrioritizeMST}\left(G, w\right)$ עפ"מ ב־G ביחס ל־w.

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

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

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

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

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 $s_1 \ldots s_n, f_1 \ldots f_n \in \mathbb{R}$ 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 $\ell\left(C\right) < 0$.

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

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 $\delta\left(s, t\right) = \inf_{\sigma \in \{s \to t\}} \ell\left(\sigma\right)$.

בעיית המסלולים הקצרים מנקודת מוצא (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־שיוויון המשולש: יהיו $u, v, w \in V$ אזי $\delta\left(u, v\right) \leq \delta\left(u, w\right) + \delta\left(w, v\right)$.

למה תת־מסלול קצר ביותר: יהי σ מסלול קצר ביותר אזי $\left(\sigma\left[i\right], \ldots \sigma\left[i + k\right]\right)$ מסלול קצר ביותר.

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

```
function BellmanFord(G, ℓ, s):
    (d, π) ← dict(V)
    d[s] ← 0
    for u ∈ V do
        d[u] ← ∞
        π[u] ← None
    end
    (c, i) ← 1
    while (i ≤ |V|) ∧ (c ≠ 0) do
        c ← 0
        for (u, v) ∈ E do
            c ← Relax(ℓ, d, u, v)
        end
    end
    return c

function Relax(ℓ, d, u, v):
    if d[v] > d[u] + ℓ(u, v) then
        d[v] ← d[u] + ℓ(u, v)
        π[v] ← u
        return 1
    end
    return 0
```

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

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

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

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 $\sigma\in\{s\to t\}$ מסלול אזי לאחר הפעלת הרצף $\text{Relax}(\sigma[0],\sigma[1]),\ldots,\text{Relax}(\sigma[n-1],\sigma[n])$ 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)$\Longleftrightarrow$(קיים מעגל שלילי אשר ניתן להגיע אליו מ־s).
- (BellmanFord החזיר 0)$\Longleftrightarrow$(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 באיזשהו 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 $\ell:E\to\mathbb{Z}$ וכן $\ell(e)\geq-W$ אזי קיים אלגוריתם לבעיית SSSP בסיבוכיות $\mathcal{O}\left(|E|\log^2(|V|)\log(|V|\cdot W)\log\log(|V|)\right)$.

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

```
function IsZeroCircle(G, ℓ):
    s ∉ V
    V ← V ∪ s
    for v ∈ V\{s} do
        E ← E ∪ {(s, v)}
        ℓ((s, v)) ← 0
    end
    BellmanFord(G, ℓ, s)
    for e ∈ E do
        if δ(s, v) ≠ δ(s, u) + ℓ(u, v) then
            E ← E\{(s, v)}
        end
    end
    if ∃ circle C ∈ 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 $\ell\left(C\right)=0$.

טענה: יהי C מעגל עבורו $\ell\left(C\right)=0$ 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)$\Longleftrightarrow$(IsZeroCircle מחזיר true).

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

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

```
function SSSP-DAG(G, ℓ, s):
    (d, π) ← dict(V)
    d[s] ← 0
    for u ∈ V do
        d[u] ← ∞
        π[u] ← None
    end
    /* Knuth'sAlgorithm is an algorithm to compute a topological sorting.  */
    f ← Knuth'sAlgorithm(G)
    for i ∈ [|V|] do
        for v ∈ Adj(f(i)) do
            Relax((f(i), v))
        end
    end
    return (d, π)
```

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

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

אלגוריתם דייקסטרה למציאת מסלולים קצרים ביותר מנקודת מוצא בגרף ללא משקלים שליליים: יהי G גרף מכוון עבורו $\ell\geq0$ ויהי $s\in V$ אזי

```
function Dijkstra(G, ℓ, s):
    Q ← heap((V, int))
    (d, π) ← dict(V)
    d[s] ← 0
    for u ∈ V do
        d[u] ← ∞
        π[u] ← None
    end
    Q.insert((s, d[s]))
    while Q ≠ ∅ do
        u ← Q.min
        for v ∈ Adj(u) do
            if d[v] = ∞ then
                π[v] ← u
                d[v] ← d[u] + ℓ(u, v)
                Q.insert((v, d[v]))
            else if d[u] + ℓ(u, v) < d[v] then
                π[v] ← u
                d[v] ← d[u] + ℓ(u, v)
                Q.decrease-key((v, d[v]))
        end
    end
    return (d, π)
```

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

משפט: יהי $s\in V$ אזי Dijkstra פתרון לבעיית SSSP כאשר $\ell\geq0$.

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

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

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

פונקציית משקל מותאמת: תהא p פונקציית פוטנציאל אזי פונקציית משקל ℓ_{p} עבורה לכל $u,v\in V$ המקיימים $\left(u,v\right)\in E$ מתקיים $\ell_{p}\left(\left(u,v\right)\right)=\ell\left(\left(u,v\right)\right)+p\left(u\right)-p\left(v\right)$.

משפט: תהא p פונקציית פוטנציאל יהיו $s,t\in V$ ויהי σ מסלול מ־s ל־t אזי $\ell_{p}\left(\sigma\right)=\ell\left(\sigma\right)+p\left(s\right)-p\left(t\right)$.

מסקנה: תהא p פונקציית פוטנציאל יהיו $s,t\in V$ ויהי σ מסלול מ־s ל־t אזי (σ מסלול קצר ביותר ביחס ל־ℓ)$\Longleftrightarrow$(מסלול קצר ביותר ביחס ל־ℓ_{p}).

מסקנה: תהא p פונקציית פוטנציאל ויהי σ מעגל אזי $\ell_{p}\left(\sigma\right)=\ell\left(\sigma\right)$.

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

פונקציית פוטנציאל פיזבילית: יהי G גרף מכוון וממושקל ℓ אזי פונקציית פוטנציאל p עבורה $\ell_{p}\geq0$.

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

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

function `FeasiblePotential`(G,ℓ)**:**

> $G' \leftarrow G \uplus \{s\}$
> **for** $v \in V(G)$ **do**
> > $E(G') \leftarrow E(G') \cup \{(s,v)\}$
> > $\ell((s,v)) \leftarrow 0$
>
> **end**
> $c \leftarrow \mathrm{BelmanForm}(G',\ell,s)$
> **if** $c=1$ **then**
> > **return** None
>
> **end**
> **else**
> > $p \leftarrow (V(G) \to \mathbb{R})$
> > **for** $v \in V(G)$ **do**
> > > $p(v) \leftarrow \delta(s,v)$
> >
> > **end**
> > **return** p
>
> **end**

טענה: יהי G גרף מכוון וממושקל ℓ אזי

- (G מצוייד עם ℓ בעל מעגל שלילי)$\iff$(FeasiblePotential (G,ℓ) מחזיר None).
- (G מצוייד עם ℓ בעל פונקציית פוטנציאל פיזבילית)$\iff$FeasiblePotential (G,ℓ) מחזיר פונקציית פוטנציאל פיזבילית).

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

function `Johnson`(G,ℓ)**:**

> $p \leftarrow \mathrm{FeasiblePotential}(G,\ell)$
> **if** $p = None$ **then**
> > **return** None
>
> **end**
> $\ell_p \leftarrow (E \to \mathbb{R})$
> **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**
> > $(d,\pi) \leftarrow \mathrm{Dijkstra}(G,\ell_p,v)$
> > /* Here D and Π will be simplified, to get a solution to APSP as needed we can modify a bit Dijkstra's algorithm to calculate D and Π 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,Π)

משפט: יהי G גרף מכוון וממושקל ℓ אזי Johnson (G,ℓ) פתרון לבעיית APSP.

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

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 $\delta_k(s,v) = \min\{\ell(\sigma) \mid (\sigma \in \{s \to v\}) \wedge (|\sigma| \le k)\}$.

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

סימון: יהי $s\in V$ אזי $\delta_k(s)\in M_{1\times|V|}(\mathbb{R})$ באשר $\left(\delta_k(s)\right)_v=\delta_k(s,v)$ 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 $D^{(k)}_{u,v}=\delta_k(u,v)$.

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: $L^k=L*\ldots*L$.

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 $\star$ פעולה אסוציאטיבית אזי

```
function RepeatedSquaring(A, ⋆):
    a_k ... a_0 ← (n)_2
    B ← M_n(ℝ)
    for i ∈ [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 $\mathcal{O}\left(|V|^3\log(|V|)\right)$.

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 $F^{(k)}_{u,v}=\min\left\{\ell(\sigma)\mid(\sigma\in\{u\to v\})\wedge(\text{למעט בהתחלה ובסוף } [k] \text{ עוברת דרך הצמתים } \sigma)\right\}$.

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 $F^{(0)}_{u,v}=L$.

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

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

```
function FloydWarshall(n, L):
    Π ← M_n([n])
    for u ∈ [n] do
        for v ∈ [n] do
            if (u ≠ v) ∧ (L_{u,v} < ∞) then
                Π_{u,v} ← u
            else
                Π_{u,v} ← None
        end
    end
    F ← L
    for k ∈ [n] do
        for u ∈ [n] do
            for v ∈ [n] do
                if F_{u,k} + F_{k,v} < F_{u,v} then
                    F_{u,v} ← F_{u,k} + F_{k,v}
                    Π_{u,v} ← Π_{k,v}
            end
        end
    end
    return (F, Π)
```

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

טענה: יהי $([n],E)$ גרף מכוון ותהא $L\in M_n(\mathbb{R})$ מטריצת המשקל אזי סיבוכיות הריצה של $\text{FloydWarshall}(n,L)$ 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 $w:[n]\to\mathbb{R}_{\ge0}$ אזי $\text{mis}(i)=\max\{w(I)\mid(I\subseteq[i])\wedge(\text{בלתי תלויה } I)\}$.

טענה: יהי $([n],E)$ גרף שרוך ויהי $w:[n]\to\mathbb{R}_{\ge0}$ 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 $w:[n]\to\mathbb{R}_{\ge0}$ 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 $f:[|B|]\to[|A|]$ עולה ממש וחח״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 $B\lhd A$.

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

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

טענה: תהיינה $A,B\in\Sigma^*$ אזי $\text{lcs}(k,\ell)=\begin{cases}0 & (k=0)\vee(\ell=0)\\ \text{lcs}(k-1,\ell-1)+1 & (k,\ell>0)\wedge(A_k=B_\ell)\\ \max\{\text{lcs}(k-1,\ell),\text{lcs}(k,\ell-1)\} & (k,\ell>0)\wedge(A_k\neq B_\ell)\end{cases}$.

מסקנה: תהיינה $A,B\in\Sigma^*$ 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 $\max\{|C|\mid(C\lhd A)\wedge(\forall i.C_{i-1}\prec C_i)\}$.

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

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

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 $k=\arg\max\{\text{lenlis}(1),\ldots,\text{lenlis}(|A|)\}$ אזי $\left(x_{\pi\text{lis}^{(\ell)}(k)},\ldots,x_{\pi\text{lis}^{(2)}(k)},x_{\pi\text{lis}(k)},x_k\right)$ פתרון של LIS בעל סיבוכיות $\mathcal{O}\left(|A|^2\right)$.

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 $(\min\text{lis}(1),\ldots,\min\text{lis}(\ell))$ פתרון של LIS בעל סיבוכיות ריצה $\mathcal{O}(|A|\cdot\log(|A|))$.

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

בעיית עץ חיפוש בינארי סטטי אופטימלי: יהיו $p_1\ldots p_n\in(0,1]$ 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 $\operatorname{costp}(T) = \left(\sum_{i=1}^{n} p_i\right) + \operatorname{costp}(T.\text{left}) + \operatorname{costp}(T.\text{right})$.

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 $\operatorname{pp}(i,j) = \sum_{k=i}^{j} p_k$.

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

טענה: יהיו $p_1 \dots p_n \in (0,1]$ ויהיו $x_1 \dots x_n$ אזי $\operatorname{cp}(i,i-1) = 0$ וכן $\operatorname{cp}(i,i) = p_i$ וכן $\operatorname{cp}(i,j) = \operatorname{pp}(i,j) + \min_{i \le k \le j}(\operatorname{cp}(i,k-1) + \operatorname{cp}(k+1,j))$.

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
```

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

מסקנה: יהיו $p_1 \dots p_n \in (0,1]$ אזי $\operatorname{OSBST}(\text{pp})$ 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 $W, w_1 \dots w_n, v_1 \dots v_n \ge 0$ אזי $S \subseteq [n]$ באשר $\sum_{i \in S} v_i$ מקסימלית וכן $\sum_{i \in S} w_i \le W$.

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

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

סימון: יהיו $W, w_1 \dots w_n, v_1 \dots v_n \ge 0$ אזי $\operatorname{bknap}(k,W) = \max\left\{\sum_{i \in S} v_i \mid (S \subseteq [k]) \wedge \left(\sum_{i \in S} w_i \le W\right)\right\}$.

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

- יהי $m \ge 0$ אזי $\operatorname{bknap}(0,m) = 0$.
- יהי $i \in [n]$ אזי $\operatorname{bknap}(i,0) = 0$.
- יהי $m \ge 0$ ויהי $i \in [n]$ אזי $\operatorname{bknap}(i,m) = \begin{cases} \operatorname{bknap}(i-1,m) & w_i > m \\ \max\{\operatorname{bknap}(i-1,m), \operatorname{bknap}(i-1,m-w_i)+v_i\} & w_i \le m \end{cases}$.

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

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

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

```
function FractionalKnapsack(W, w_1, ..., w_n, v_1, ..., v_n):
    f ← ([n] → [0, 1])
    P ← List([n] × ℝ)
    for i ← [n] do
        P(i) ← (i, v_i / w_i)
        f(i) ← 0
    end
    P ← sort(P) // Sort from high to low based on second coordinate.
    t ← 0
    i ← 1
    while (t < W) ∧ (i ≤ n) do
        j ← P(i)[0]
        if t + w_j ≤ W then
            f(j) ← 1
            t ← t + w_j
        end
        else
            f(j) ← (W − t) / w_j
            t ← W
        end
    end
    return f
```

```
function ZeroOneKnapsack(W, w_1, ..., w_n, v_1, ..., v_n):
    k ← n
    w ← W
    S ← Set([n])
    S ← ∅
    while (k > 0) ∧ (w > 0) do
        if bknap(k, w) ≠ bknap(k − 1, w) then
            S ← S ∪ {k}
            k ← k − 1
            w ← w − w_k
        else
            k ← k − 1
    end
```

רשת זרימה: יהי G גרף מכוון וממושקל $c \geq 0$ ותהיינה $s,t \in V$ 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 $f : E \to \mathbb{R}_{\geq 0}$ עבורה

- חסם קיבולת: $f \leq c$.
- שימור זרם: לכל $v \in V \backslash \{s,t\}$ מתקיים $\sum_{\substack{u \in V \\ (v,u) \in E}} f\left((v,u)\right) = \sum_{\substack{u \in V \\ (u,v) \in E}} f\left((u,v)\right)$.

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

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

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

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

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

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

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

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

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

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

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

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

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

טענה הגדלת מסלול: תהא f זרימה ויהי $P \in \{s \to t\}$ מסלול ניתן להגדלה s-t אזי קיימת פונקציית זרימה g עבורה $g_{\restriction_{E \backslash P}} = f_{\restriction_{E \backslash P}}$ וכן $|f| < |g|$.

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

- $E_f = \{e \in E \mid c\left(e\right) > f\left(e\right)\} \cup E^{-1}$.
- פונקציית שיוריות הקיבולת: תהא $e \in E_f$ אזי $c_f\left(e\right) = \begin{cases} c(e)-f(e) & e \in E \\ f\left(e^{-1}\right) & e \in E^{-1} \end{cases}$.

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

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

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

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

משפט: תהא f זרימה התב"ש

- f זרימה מקסימלית ב־G.
- לכל מסלול $P \in \{s \to t\}$ 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 $\text{FF} = \text{FordFulkerson}\left(V,E,c,s,t\right)$.

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

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

- f זרימה של G.
- $f\left(E\right) \subseteq \mathbb{N}$.
- $c_f\left(P\right) \geq 1$.

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

```
function FordFulkerson(V, E, c, s, t):
    f ← (E → ℝ)
    f ← 0
    while True do
        G_f ← ResidualNetwork(G, c, s, t, f) // Construct it like any graph.
        π_{G_f} ← BFS(G, s)
        if {s → t} ∩ π_{G_f} = ∅ then
            return f
        else
            P ← {s → t} ∩ π_{G_f} // The path is taken from π_{G_f}.
            f ← f_P
    end
```

- FF פתרון לבעיית הזרימה המקסימלית לכל בחירת מסלולים באלגוריתם.
- FF עושה לכל היותר $|f|$ שיפורי מסלול.
- $\mathbb{N} \subseteq (E)$ FF.

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

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