גרף מכוון קשיר: גרף מכוון G עבורו לכל $u,v\in V\left(G\right)$ קיים מסלול מ־u ל־v או מסלול מ־v ל־u.

גרף מכוון קשיר חזק: גרף מכוון G עבורו לכל $u,v\in V\left(G\right)$ קיים מסלול מ־u ל־v.

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

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

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

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

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

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

משפט נכונות מרחקים: יהי G גרף ויהיו $s,v\in V$ אזי $d\left[v\right]=\delta\left(v,s\right)$.$\text{BFS}\left(G,s\right)$.

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

טענה: יהי G גרף ויהי $s\in V$ אזי

- מתקיים $\deg_{G_{\pi}}^{-}\left(s\right)=0$.
- לכל $v\in V\left(G_{\pi}\right)$ מתקיים $\deg_{G_{\pi}}^{-}\left(v\right)=1$.
- לכל $v\in V\left(G_{\pi}\right)$ קיים מסלול ב־G_{π} בין s,v.
- G_{π} הינו עץ.
- יהי $v\in V\left(G_{\pi}\right)$ 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 $\deg\left(v\right)\in\mathbb{N}_{\text{even}}$).

אלגוריתם למציאת מעגל אוילר: יהי G גרף קשיר ולא מכוון עבורו לכל $u\in V$ מתקיים $\deg\left(u\right)\in\mathbb{N}_{\text{even}}$ 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 $\text{EulerCircle}(G, v)$ 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 $\text{EulerCircle}(G)$ הינו מעגל אוילר.

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

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)$\Longleftrightarrow$(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)$\Longleftrightarrow$($\text{IsBipartite}(G) = \text{true}$).

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

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)$\Longleftrightarrow$(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 $E' = \{e \in E \mid$ e חלק ממסלול קצר ביותר היוצא מ־s ל־$t\}$ 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 $v \in [s]_{\rightarrow}$ 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 $\mathrm{DFS}(G)$.

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 $[k(u), f(u)] \cap [k(v), f(v)] = \varnothing$ 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)$\iff$(בזמן $k(u)$ באלגוריתם $\mathrm{DFS}(G)$ 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 $\mathcal{O}(|V|+|E|)$.

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

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

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

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 $\text{DFS}(G)$.

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 |Adj_{G_π}(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 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, 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(G^T)
    A ← set(set(V))
    for v ∈ V do
        A.append([v]_{→ G_π^T})
    end
    return A
```

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

אלגוריתם למציאת גרף הרכיבים: יהי G גרף מכוון אזי

```
function ComponentGraph(G):
    V* ← SCC(G)
    E* ← set((V*)^2)
    for (u, v) ∈ E do
        if [v]_{G_π^T} ≠ [u]_{G_π^T} then
            E*.append(([v]_{G_π^T}, [u]_{G_π^T}))
        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 ← set(V(G))
    G* ← ComponentGraph(G)
    for C ∈ V(G*) do
        v ← {u ∈ C | ∄w ∈ V(G)\C.(w, u) ∈ 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 $w : E \to \mathbb{R}$ 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 $w(T) = \min\{w(S) \mid G \text{ עץ פורש של } S\}$.

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

קשתות החתך: 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 $G \leq T$ 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 $V(G) \setminus [v]_{\xrightarrow[T-\{e\}]{}}, [v]_{\xrightarrow[T-\{e\}]{}}$ 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 $\mathcal{O}\left(E\log\left(V\right)\right)$.

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

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

function `Kruskal'sAlgorithm`(G):
  color $\leftarrow$ dict(E)
  $L \leftarrow$ sort(E)
  for $(u,v) \in L$ **do**
    if $\exists\sigma \in \{u \to v\}.\forall i \in [n].color(\sigma(i)) = Blue$ **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 $\text{Kruskal'sAlgorithm}\left(G\right)$ 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 $\mathcal{O}\left(E\log\left(V\right)\right)$ וכן סיבוכיות amortized $\mathcal{O}\left(E\log^{*}\left(V\right)\right)$.

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

function `Borůvska'sAlgorithm`(G):
  Trees $\leftarrow$ set(set(G))
  for $v \in V$ **do**
    Trees.append($\{v\}$)
  end
  while $|Trees| \neq 1$ **do**
    for $T \in Tree$ **do**
      $(u,v) \leftarrow \text{argmin}_{(u,v)\in V(T)\times V(G)}(w((u,v)))$
      $S \leftarrow \{S \in Tree \mid u \in V(S)\}$
      $S \leftarrow S + T + \{(u,v)\}$
      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 $\text{Borůvska'sAlgorithm}\left(G\right)$ הינה $\mathcal{O}\left(E\log\left(V\right)\right)$.

משפט: יהי G גרף קשיר לא מכוון וממושקל w באשר w חח"ע אזי קיים ויחיד $T \leq G$ 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 $T \leq G$ עבורו $A \subseteq E\left(T\right)$ וכן $e \notin E\left(T\right)$.