

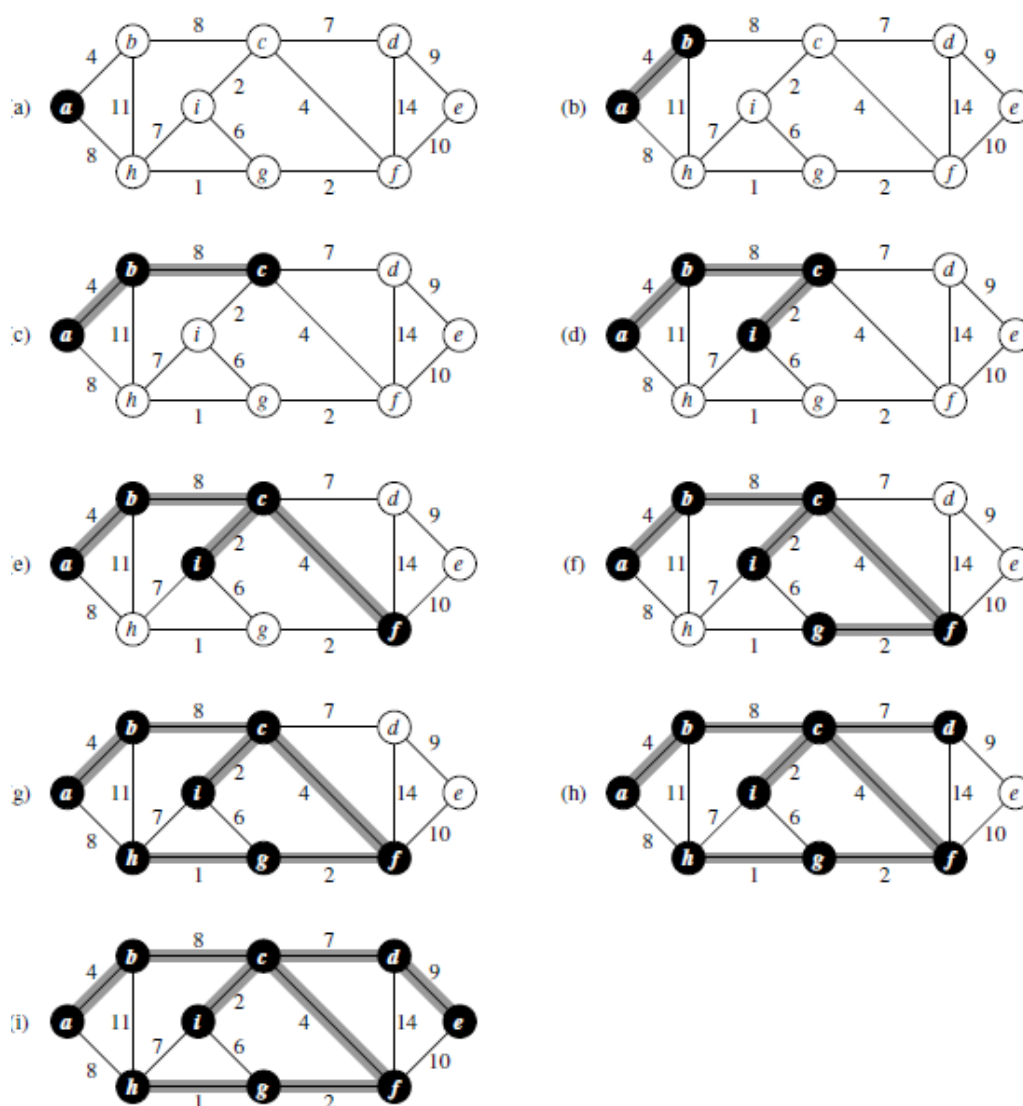
## Minimum Spanning Tree - Prim Algorithm

**האלגוריתם של פריים** הוא אלגוריתם **חמדני** המשמש למציאת עץ פורש מינימלי בגרף משוקלל לא מכוון. האלגוריתם מתחיל את בניית העץ מקדקוד פתיחה שנבחר באקראי. בכל צעד האלגוריתם מוסיף לעץ את הצלע בעלת המשקל המינימלי מבין אלה היוצאות מקדקודי העץ ולא סוגרות מעגל, יותר פורמאלי:  
 1. בוחרים בקראי קדקוד כלשהו  $s$  ומוסיפים אותו לעץ:

Initialization:  $T = \{s\}$

2. כול עוד  $T$  מכיל פחות מ- $n-1$  קודקודים מוסיפים ל- $T$  צלע בעלת משקל מינימלי, כך שבדיוק קדקוד אחד נמצא ב- $T$  וקדקוד אחר לא ב- $T$ .

$T = T \cup \{a,b\} : ((a \in T \text{ and } b \notin T) \text{ or } (a \notin T \text{ and } b \in T)) \text{ and } \text{weight}(a,b) \rightarrow \min$



שליבים של האלגוריתם בהתאם לדוגמא:

סימונים:

משבצת אפורה – קדקוד עזב את התור Q.

**אות אדומה** – מצב הקדקוד השתנה בהשוואה לשלב הקודם.

שלב 1

key	P	Q	קדקודים
0	NIL		a
4	a	b	b
$\infty$	NIL	c	c
$\infty$	NIL	d	d
$\infty$	NIL	e	e
$\infty$	NIL	f	f
$\infty$	NIL	g	g
8	a	h	h
$\infty$	NIL	i	i

אתחול

key	P	Q	קדקודים
0	NIL	a	a
$\infty$	NIL	b	b
$\infty$	NIL	c	c
$\infty$	NIL	d	d
$\infty$	NIL	e	e
$\infty$	NIL	f	f
$\infty$	NIL	g	g
$\infty$	NIL	h	h
$\infty$	NIL	i	i

שלב 3

key	P	Q	קדקודים
0	NIL		a
4	a		b
8	b		c
7	c	d	d
$\infty$	NIL	e	e
4	c	f	f
$\infty$	NIL	g	g
8	a	h	h
2	c	i	i

שלב 2

key	P	Q	קדקודים
0	NIL		a
4	a		b
8	b	c	c
$\infty$	NIL	d	d
$\infty$	NIL	e	e
$\infty$	NIL	f	f
$\infty$	NIL	g	g
8	a	h	h
$\infty$	NIL	i	I

שלב 5

key	P	Q	קדקודים
0	NIL		a
4	a		b
8	b		c
7	c	d	d
10	f	e	e
4	c		f
2	f	g	g
7	i	h	h
2	c		i

שלב 4

key	P	Q	קדקודים
0	NIL		a
4	a		b
8	b		c
7	c	d	d
$\infty$	NIL	e	e
4	c	f	f
6	i	g	g
7	i	h	h
2	c		I

שלב 7

key	P	Q	קדקודים
0	NIL		a
4	a		b
8	b		c

שלב 6

key	P	Q	קדקודים
0	NIL		a
4	a		b
8	b		c

7	c	d	d
10	f	e	e
4	c		f
2	f		g
1	g		h
2	c		i

שלב 9

key	P	Q	קדקודים
0	NIL		a
4	a		b
8	b		c
7	c		d
9	d		e
4	c		f
2	f		g
1	g		h
2	c		i

7	c	d	d
10	f	e	e
4	c		f
2	f		g
1	g	h	h
2	c		i

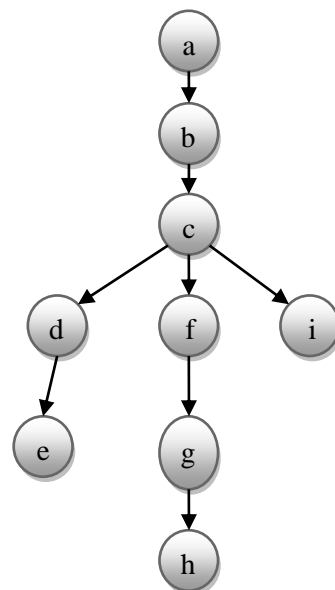
שלב 8

key	P	Q	קדקודים
0	NIL		A
4	a		B
8	b		C
7	c		D
9	d	e	E
4	c		F
2	f		G
1	g		H
2	c		i

### התוצאה:

קבלנו עץ פורש מינימאלי שמשקלו 37 (ניתן לבדוק לפי האירור או לפי הסכום של שדות key).

לפי מערך P של קדקודי אבות (parent vertices) ניתן לבנות עץ:



## Prim pseudo code

```
Prim(G, root)
  Edge T[n-1] <- empty tree (array of edges)
  numEdges = 0
  for each v in V(G, root) //O(n)
    visited[v] = false
    key[v] = infinity
    parent(v) = NIL
  end-for
  key[root] = 0
  Q <- V(G) // Q Min Heap, Q keyed by key[v] //O(n)
  while (Q != empty && numEdges < n-1)
    u = extractMin(Q) //O(log2(n))
    for each v in Adj(u)
      if (visited[v] == false && key[v] > weight(u,v))
        key[v] = weight(u,v)
        parent[v] = u
        decreaseKey(Q, v, weight(u,v)) //O(log2(n))
      end-if
    end-for
    visited[u] = true
    x = Get-Min(Q)
    T[numEdges++] = (parent[x], x)
  end-while
end-Prim
```

## סיבוכיות:

בעת מימוש האלגוריתם נעשה שימוש בערימה שמתוכה מוציאים בכל פעם את הצלע המינימלית. אם משתמשים בערימה בינארית (heap) סיבוכיות האלגוריתם תהיה  $O(|E| \log_2 |V|)$ , (כאשר  $|E|$  הוא מספר הצלעות ו- $|V|$  הוא מספר הקדקודים).

באופן כללי היעילות של האלגוריתם של פריים טובה מזו של האלגוריתם של קרוסקל. למרות זאת, אם הקלט כבר ממוין לפי משקלי הצלעות או כאשר ניתן למיין אותם בזמן לינארי, אזי האלגוריתם של קרוסקל יהיה מהיר יותר. הסיבוכיות של אלגוריתם של קרוסקל ללא מיון היא  $O(|V| \log_2 |V|)$ .

הסיבוכיות של אלגוריתם של קרוסקל עם מיון היא מורכבת מסיבוכיות של מיון  $O(|V| \log_2 |V|)$  וסיבוכיות של אלגוריתם עצמו  $O(|E| \log_2 |V|)$   
 $O(|V|) + O(|V| \log_2 |V|) + O(|E| \log_2 |V|) = O(|E| \log_2 |V|)$

## הוכחת נכונות של אלגוריתם של פרים.

נוכיח שבכל שלב שאנו מוסיפים צלע חדשה ל- $T$  אנו מקבלים תת-עץ של עץ פורש מינימאלי כלשהו  $M$ .

### הוכחה באינדוקציה.

(א) בסיס. בשלב ראשון של האלגוריתם אנו מוציאים מתור עדיפויות  $Q$  ( $\min$  heap) את שורש העץ ( $\text{root}$ ), שהוא שייך לכל עץ פורש מינימאלי.

(ב) נניח שבשלב כלשהו של פרים קבלנו  $T$  תת-עץ של  $M$ .

(ג) בשלב הבא של פרים אנו מוסיפים צלע  $e$  ל- $T$ . אם  $e \in M$  אז גם  $T \cup \{e = (a, b)\} \subseteq M$  ובזה סיימנו את ההוכחה. נניח ש- $e \notin M$ . נוסיף  $e$  ל- $M$ , נקבל מעגל. במעגל זה יש מסלול שמחבר את קדקוד  $a$  עם  $b$  וכוון שרק קדקוד אחד של  $e$  שייך ל- $T$ , (כי אלגוריתם של פרים מוסיף את  $e$ ) במעגל זה קיימת צלע  $g = (x, y)$  כלשהי שרק קדקוד אחד שלה שייך ל- $T$ . אלגוריתם של פרים היה יכול להוסיף  $e'$  ל- $T$ , אבל הוא בחר ב- $e$ , כלומר  $\text{weight}(e) \leq \text{weight}(g)$ . נבנה עץ חדש  $M' = M \cup \{e\} - \{g\}$  שמשקלו קטן או שווה למשקל של  $M$ . אבל  $M$  הוא בעל משקל מינימאלי, לכן  $\text{weight}(M) = \text{weight}(M')$  ו- $M'$  הוא גם עץ פורש מינימאלי שמכיל את  $T$ :  $T \cup \{e\} \subseteq M'$ . מש"ל.