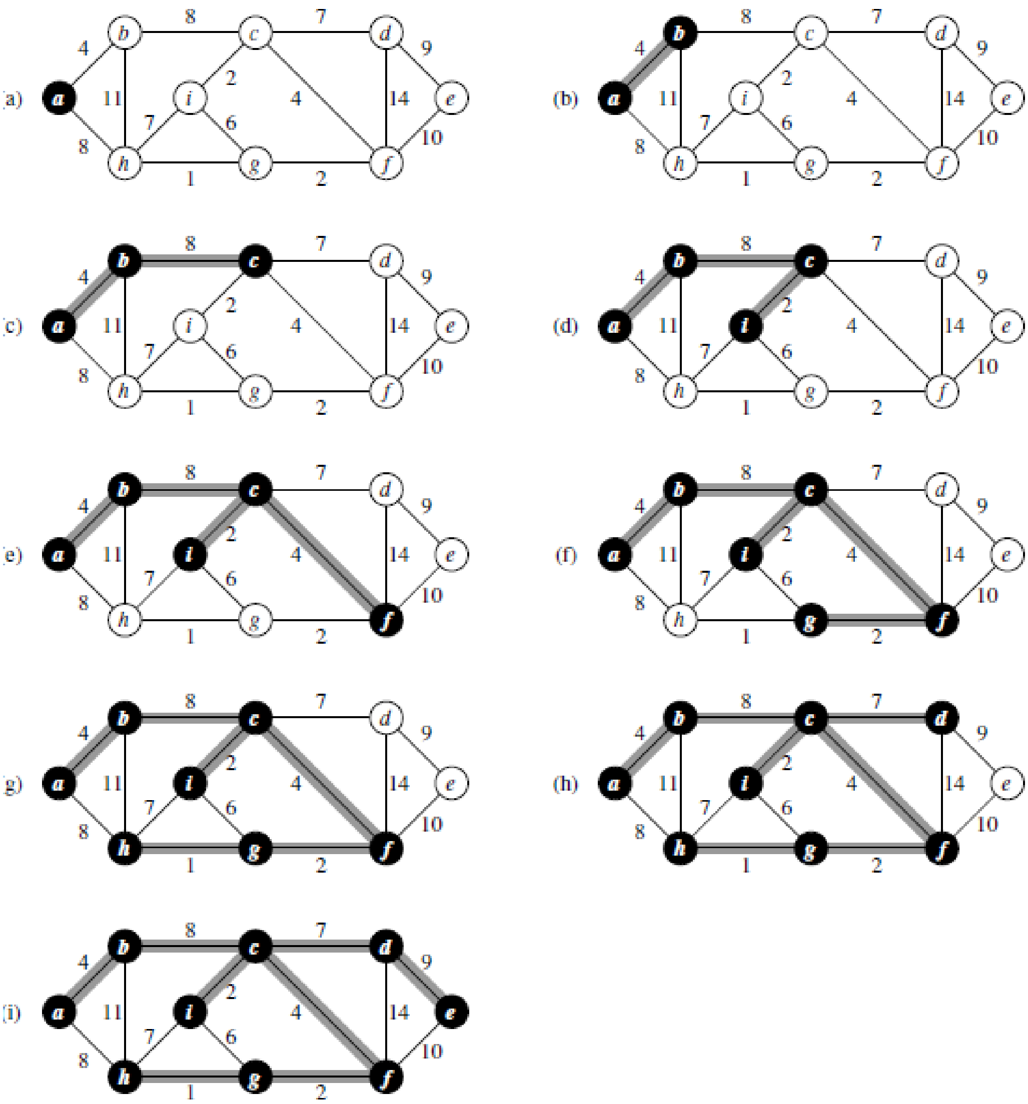
Minimum Spanning Tree - Prim Algorithm

האלגוריתם של פרים 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.

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

באופן כללי היעילות של האלגוריתם של פרים 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.



Prim algorithm for Minimum Spanning Tree problem

G – graph
w – weights of edges
r – root
Q – priority queue (heap) based on a *key* field. For each vertex *v*, *key[v]* is the minimum weight of any edge connecting *v* to a vertex in a tree; *key[v]*$=\infty$ if there is no such edge. The field *p[v]* names the parent of v in the tree.
NIL = -1 inexistent index

```
Prim(G, w, r)
    for each u∈V[G]
        key[u] = ∞ // set the key of each vertex to ∞ (except for the root r)
        p[u] = NIL // the parent of each vertex is NIL = -1
    end for
    key[r] = 0 // the first vertex proceeded
    Q←V[G] // the min-priority queue Q contains all the vertices
    while(Q is not empty)
        // extract vertex with min weight (key[u] – min)   from Q
        u = EXTRACT-MIN(Q)
        for each v∈Adj[u] // for each v 2eighbor of u
            if (v∈Q and w(u,v)<key[v] then
                key[v] = w(u,v) //the weight of v is the weight of edge (u,v)
                p[u] = v // the parent of v is u
            end if
        end for
    end while
end Prim
```

שלבים של האלגוריתם בהתאם לדוגמה:

אתחול

key	P	Q	קדקודים
0	NIL	a	a
∞	NIL	b	b
∞	NIL	c	c
∞	NIL	d	d
∞	NIL	e	e
∞	NIL	f	f
∞	NIL	g	g
∞	NIL	h	h
∞	NIL	i	i

שלב 1

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

<u>סימונים:</u>

משבצת אפורה – קדקוד עזב את התור Q.

אות אדומה – מצב הקדקוד השתנה בהשוואה לשלב הקודם.

שלב 2

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

שלב 3

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

שלב 4

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

שלב 5

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

שלב 6

קדקודים	Q	P	key
a		NIL	0
b		a	4
c		b	8
d	d	c	7
e	e	f	10
f		c	4
g		f	2
h	h	g	1
i		c	2

שלב 7

קדקודים	Q	P	key
a		NIL	0
b		a	4
c		b	8
d	d	c	7
e	e	f	10
f		c	4
g		f	2
h		g	1
i		c	2

שלב 8

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

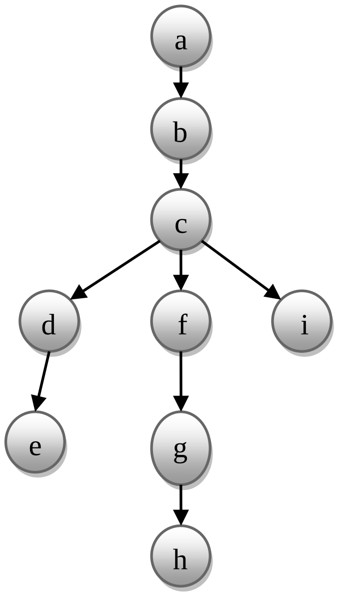
שלב 9

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

התוצאה:

קבלנו עץ פורש מינימאלי שמשקלו 37 (ניתן לבדוק לפי האיור 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)
          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]
      while (Q != empty && numEdges<n-1)
          u = Extract-Min(Q)
          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))
              end-if
           end-for
           visited[u] = true
           x=Get-Min(Q)
           T[numEdges++]=(parent[x], x)
      end-while

end-Prim
```

סיבוכיות:

$$O(|V|) + O(|E|\log_2(|V|)) = O(|E|\log_2(|V|))$$

הוכחת נכונות של אלגוריתם פרים.

נוכיח שבל שלב שאנו מוסיפים צלע חדשה ל-T אנו מקבלים תת-עץ של עץ פורש מינימאלי.

הוכחה באנדוקציה.

א) בסיס. בשלב ראשון של האלגוריתם אנו מוציאים מתור עדיפויות Q (min heap) את שורש העץ (root) ומוסיפים לעץ פורש מינימאלי T צלע שיוצאת משורש ובעלת משקל מינימאלי.

ב) נניח שבשלב כלשהו קבלנו T_1 תת-עץ של T.

ג) לפני שמוציאים קדקוד חדש מ-Q אנו קבלנו שתי קבוצות זרות של קדקודי הגרף: קבוצת קדקודים ששייכים ל- T_1 וקדקודים ששייכים לקבוצה $V-T_1$. נוכיח שצלע חדשה $e=(a,b)$ בעלת משקל מינימאלי שמוסיפים אותה ל-T_1 יוצרת $T_1\cup\{e\}$ תת-עץ של עץ פורש מינימאלי. נשלים את T_1 עד עץ פורש מינימאלי כלשהו ונסמן אותו T_{min}. אם $e\in T_{min}$

הוכחנו את הטענה. נניח ש- $e \notin T_{min}$. נתבונן במסלול P ב-T_{min} שמחבר קדקודים a ו-b. צלע e מחברת קדקודים ששייכים לקבוצות זרות (קדקודי T_1 וקדקודי $V-T_1$). לכן במסלול P קיימת צלע e_1 שמחברת שתי הקבוצות האלה ו- $weight(e) \le weight(e_1)$. נחליף ב-T_{min} צלע e_1 בצלע e , נקבל

$$T_2 = T_{min} - \{e_1\} + e$$

שהוא גם עץ פורש מינימאלי של G כי יש בו אותו מספר צלעות כמו ב- T_{min}, הוא קשיר ומשקלו לא עולה על משקל של T_{min}. אז ניתן להשלים את $T_1 \cup \{e\}$ עד עץ פורש מינימאלי שהוא T_2. מש"ל.