אלגוריתם Prim:

מטרה: ליצור עץ פורש מינימאלי.

רעיון: מתחילים מקודקוד כלשהו על הגרף (כי בסוף צריך את כל הקודקודים) וממנו בונים את העץ באופן הבא, לוקחים את הצלע בעלת המשקל הכי קטן שמחוברת לקודקוד שעומדים עליו ומגיעים לעוד קודקוד.

כעת, יש 2 קודקודים בקבוצה שביקרנו בהם, ניקח את הצלע בעלת המשקל הכי קטן שמחוברת לאחד מקודקודי הקבוצה שבה ביקרנו.  
איך דואגים לא לסגור מעגל? בודקים לפני שלוקחים את הצלע, האם כבר ביקרנו בקודקוד שבצד השני של הצלע. (אם כן - הצלע סוגרת מעגל ולא ניקח אותה)

איך נגע לחפש תמיד את הצלע המינימאלית? נשתמש בתור עדיפויות (minHeap)

האלגוריתם:

בהינתן גרף G - רשימת שכנויות. .

1. - אתחול.
2. לכל
   1. . (משקל הצלע הכי "טובה" שמחוברת ל )
3. נבחר את קודקוד 0 באופן שרירותי ונקבע: .
4. אתחל תור עדיפויות והכנס את לתור.
5. כל עוד התור לא ריק וגם
   1. הוצא את המינימאלי מהתור ושמור אותו ב .
   2. עבור כל שכן של כאשר היא הצלע ביניהם.
      1. אם וגם
         1. עדכן בתור: .
         2. עדכן:
         3. הכנס את לתור.
   3. אם התור לא ריק, שמור את המינימאלי שנמצא בתור כעת ב
6. אם החזר שאין עץ פורש
7. אחרת, החזר את .

סיבוכיות: סה"כ:

אתחול המערכים: .

אתחול התור: .

נשים לב שכל קודקוד ייכנס ויצא לכל היותר פעם אחת מהתור לכן הוצאת המינימום מהתור תהיה לכל היותר: .  
עבור כל קודקוד - עוברים על כל השכנים שלו ולכן סה"כ נבצע את הפעולות הפנימיות (הפעולות בלולאה הפנימית ביותר): . עבור הפעולות הפנימיות - הכנסה ועדכון של התור הוא: כי בתור יש את הקודקודים של הגרף. סה"כ: שאר הפעולות הן ב .