אלגוריתם בורובקה-סולין:

מטרה: ליצור עץ פורש מינימאלי.

רעיון: להתחיל ללא צלעות אלא יער של קודקודים (יער= גרף ללא מעגלים, לא בהכרח קשיר)  
בכל שלב, לזהות עבור כל עץ (בהתחלה כל קודקוד הוא עץ בפני עצמו) את הצלע הסמוכה אליו (שעוד לא נלקחה בשלבים קודמים) בעלת המשקל הנמוך ביותר. בסיום כל שלב מצרפים את כל הצלעות הנמוכות לעץ הפורש הכללי (התשובה) - במקרה כזה, יאוחדו מספר עצים לעץ אחד, חוזרים על התהליך עד שהגרף כולו הוא עץ אחד.

לצורך הבדיקה האם 2 קודקודים כבר נמצאים בעץ אחד, נשתמש ב union-find האומר האם 2 קודקודים הם באותה קבוצה.

האלגוריתם:

בהינתן גרף קשיר לא מכוון הנתון כרשימת שכנויות, - מספר הקודקודים.

1. הגדר אובייקט חדש בשם בגודל .
2. הגדר - התשובה (העץ שנחזיר בסוף).
3. הגדר מערך המכיל את כל צלעות הגרף G.
4. כל עוד
   1. הגדר מערך צלעות המאותחל ב . (במערך במיקום ה i נשים את הצלע עם המשקל הקטן ביותר שמחוברת לעץ שהתחיל מקודקוד i)
   2. עבור כל צלע
      1. . (שמרנו את קצוות הצלע ואת משקלה)
      2. אם:
         1. אם או אז:
         2. אם או אז:
   3. עבור מ 1 עד n:
      1. סמן:
      2. אם אז
5. החזר את .

סיבוכיות:

אתחול אובייקט UF הוא ואתחול רשימת הצלעות F היא

בניית מערך הצלעות E הוא:

נשים לב שבכל איטרציה של הלולאה, מחברים רק קודקודים שהיו בעצים שונים בשלב הקודם. לכן, עובר כל 2 קודקודים מעצים שונים, או שהם יחוברו ביניהם ואז הפכנו 2 עצים לאחד, או שכל אחד מהם יחובר לקודקוד אחר ואז מ 4 עצים ירדנו ל 2 או ששהם יחוברו לעץ משותף ואז מ 3 עצים ירדנו 1. בכל מקרה, כמות העצים יורדת פי 2. לכן הלולאה החיצונית תתבצע פעמים.

הלולאה שעוברת על כל הצלעות תבצע פעולות כיוון שעוברים על כל צלע ובודקים באמצעות UF אם קודקודי הקצה שלה באותה קבוצה או לא.

הלולאה שעוברת על cheepest מבצעת פעולות כיוון שעוברים על כל תא ומאחדים את קודקודי הקצה של כל צלע שנבחרה בשלב הזה.

סה"כ:   
המעבר האחרון הוא כי הגרף קשיר.