<u>חישוביות וסיבוכיות – תרגיל 1</u>

1) **הפרכה,** נגדיר את קלט הבעיה כך - $P=\{<M,x>\}$ כך שהמכונה M עוצרת על הקלט x.

נשים לב כי זו אומנם בעיית הכרעה אך ניתן להגדיר את בעיית החיפוש להיות – מצא את התשובה לבעיית ההכרעה (מכיוון שזוהי בעיה קשה יותר)

נוכיח כי $P\in PA$ - נשים לב כי תמיד נוכל להחזיר מכונה שעוצרת על כל אחד מהקלטים האפשריים.

נוכיח כי $P\notin PF$ - על מנת מנת לדעת אם אכן המכונה והקלט שייכים לP צריך לבדוק אם המכונה עוצרת על x וזה אומר (=שקול) לפתרון בעיית העצירה שראינו שנה שעברה כי היא אינה כריעה.

ולכן מכיוון שהראינו P כך ש $P\in PA$ אך גם $P\notin PF$ בהכרח $PA\neq PF$

2)א) **הוכחה,** נחלק למקרים – נניח כי $x\in S_1\cup S_2$ אזי בהכרח $x\in S_1$ ו/או $x\in S_2$ נניח בלי הגבלת כלליות כי $x\in S_2$ ולכן בהכרח קיים y כך ש $V_2(x,y)=1$ ולכן כאשר המוודא עבור $S_1\cup S_2$ ירוץ על (x,y) הוא יבדוק את $V_2(x,y)$ יקבל 1 ולכן יחזיר גם כן 1.

כעת נניח כי $x\notin S_1\cup S_2$ לכן $x\notin S_1$ וגם $x\notin S_2$ ולכן כאשר המוודא יכניס קלט (x,y) עבור כל אחד מהמוודאים הם יחזירו 0 לכל y ולכן גם המוודא של $S_1\cup S_2$ יחזיר 0

2)ב) **הפרכה,** יכול להתקיים מצב שבו קיים x ששייך ל S_1 ול S_2 זאת אומרת $x\in S_1\cap S_2$ אך הפתרון עבור כל אחת מהבעיות שונה זאת אומרת y_1 הינו פתרון עבור S_1 ו y_2 פתרון עבור S_2 אך $y_1\neq y_2$ אזי עבור המוודא הפולינומי עבור $S_1\cap S_2$ נקבל 1 כי קיים פתרון לשניהם, אך לא y יחיד שהוא הפתרון בשני המקרים.

לדוגמה - נניח כי S_1 הינה בעיית הכפל (2 מספרים וצריך לחפש את הכפל ביניהם) וכך גם S_2 הינה בעיית החיבור, לכן נשים לב כי עבור x כלשהו (2 מספרים כלשהם) בוודאי ש $x\in S_1$ וגם $x\in S_2$ ולכן $x\in S_1\cap S_2$ אך נשים לב כי אם ניקח y תשובה עבור x בבעיית החיבור היא לא בהכרח תהיה התשובה גם של בעיית הכפל ולכן $V_1(x,\mathrm{y})=0$ אך $V_2(x,\mathrm{y})=1$

(3

נבצע את האלגוריתם הבא:

נשלח את הגרף שלנו אל האורקל, ונבדוק אם הוא מחזיר כן (קיימת תשובה) נמשיך בלולאה, אחרת נחזיר לא.

<u>תחילת הלולאה –</u>

נשלח את הגרף לאורקל ובכל פעם נוותר על קשת כלשהי.

אם האורקל החזיר **כן** (קיימת תשובה) – נוריד לצמיתות את הקשת הזאת ונחזור לתחילת הלולאה.

אם האורקל החזיר **לא** (לא קיימת תשובה) – נחזיר חזרה את הקשת שהורדנו ונעבור לקשת הבאה.

לאחר סיום הלולאה, נשאר עם K הקשתות הכרחיות עבור החתך (בלי הקשתות הפנימיות) – מכיוון שאם היא הייתה הכרחית לפתרון האורקל היה מחזיר לא והיינו משאירים אותה, ואם היא לא הכרחית היינו עוברים עליה והאורקל היה מחזיר כן ולכן יכולנו להוריד אותה לצמיתות.

כעת נשאר למצוא את החתך עצמו –

נחלק את הגרף לרכיבי קשירות ובהכרח נשאר עם K רכיבים. עבור כל רכיב נוכל למצוא את החיתוך המתאים ישר כי אסור שיהיו רשתות בין 2 קודקודים בתוכו כי אם כן הן היו יורדות בשלב הקודם.
כעת נבחר קודקוד כלשהו ונשים אותו באחד הצדדים ובהכרח לשאר הקודקודים יש סידור בודד, כל השכנים לקודקוד זה יהיו בצד השני של החיתוך והשכנים של שכניו יהיו באותו צד איתו.

בסך הכל הפעלנו את האורקל V פעמים כמספר הקודקודים ומציאת החיתוך לקח K שזהו זמן פולונומיאלי.

4)

נקבל כקלט של שאלה לבעיית VC כעת על מנת לממש את הרדוקציה עבור הגרף (והk) הנתון נוסיף זוג קודקודים ונחבר ביניהם קשת ונשלח אל Almost-VC

- כעת אם היה פתרון לבעיית VC אזי בהכרח גם יהיה פתרון לבעיית Almost-VC מכיוון שהקשת שאינה מכוסה תהיה הקשת שהוספנו והפתרון יישאר הפתרון של VC.

- אם לא היה פתרון לVC אז בהכרח גם אין פתרון לAlmost-VC, נניח שיש פתרון ונגיע לסתירה –
  - נניח שהפתרון כולל את אחד מהקודקודים של הקשת החדשה שהוספנו סימן שאת שאר הגרף (חוץ מקשת בודדה) אפשר לבחור בk-1 קודקודים וזה סתירה לכך שלVC אין פתרון מכיוון שאז יכולנו לבחור את כל הקשתות חוץ מקשת אחת בk-1 קודקודים ואת הקשת האחרונה לבחור בקודקוד שנשאר להשלים לk.

  - נניח שהוא אינו כולל את אחד מהקודקודים של הקשת החדשה אזי היא הקשת שלא מכוסה, ולכן כל שאר הקשתות (שהן הקשתות של הגרף המקורי) יהיו חייבות להיות מכוסות, והן יהיו מכוסות על ידי k קודקודים בסתירה לכך שלא ניתן לכסות אותן ע"י k קודוקדים כי אין פתרון לVC.

5)

נקבל כקלט שאלה של בעיית VC וכעת על מנת לממש את הרדוקציה עבור הגרף (והk) הנתון ניקח $|V|-k$ קודקודים בלי קשתות ונשלח את הבעיה הזאת אל בעיית ה $SubGraph$ כאשר הגרף המקורי הוא G והקודקודים שייצרנו יהוו את H.

נוכיח זאת -

- אם **יש** פתרון לVC בהכרח קיים פתרון לבעיית $SubGraph$ מכיוון שאם קיים פתרון לVC אזי בהכרח קיימת קבוצה בתל בגודל $|V|-k$ מכיוון שאם לא קיימת קבוצה כזאת אז לא נוכל לבחור k קודקודים שיכסו לנו את כל הגרף. $SubGraph$ יחזיר כן עבור הקלט הנ"ל מכיוון שעל פי מה שראינו קיימת קבוצה בתל והתת גרף שלנו לבדיקה הוא מספר קודקודים שאינם מחוברים אחד לשני משמע הם בתל ולכן מכיוון שהראינו שקיימת קבוצה בגודל כזה אזי $SubGraph$ יחזיר כן.
- אם **אין** פתרון לVC בהכרח גם אין פתרון לבעיית ה $SubGraph$ מכיוון שאם אין פתרון לVC אזי לא קיימת קבוצה בתל בגודל k בגרף המקורי כי אם הייתה קיימת היינו יכולים לכסות את כל הקשתות בk קודקודים ולכן מכיוון שלא קיימת קבוצה בתל בגודל k הבעיה של $SubGraph$ שלנו תחזיר גם כי לא קיים פתרון כי לא נוכל למצוא מספר קודקודים שאין ביניהם שום קשת כמו בH שהרכבנו.