הגדרות:

גרף פשוט: גרף ללא לולאות עצמיות וקשתות מקבילות

מסלול פשוט: מסלול שבו כל צומת מופיעה לכל היותר פעם אחת.

מעגל: מסלול פשוט שבו צומת ההתחלה היא צומת הסיום.

מסלול אוילר: מסלול שעובר דרך כל הקשתות בלי לעבור באף קשת פעמיים.

מסלול המילטוני: מסלול פשוט העובר בכל הצמתים, בו מופיע כל צומת פעם אחת בדיוק.

עץ: גרף פשוט, קשיר, וחסר מעגלים.

יער: גרף חסר מעגלים אך לא בהכרח קשיר.

צומת הפרדה: 1. צומת אשר הסרתה מהגרף תגרום לו להיות לא קשיר.

2. צומת v תהיה צומת הפרדה אמ"מ קיימים צמתים a,b שכל המסלולים בינהם עוברים דרך צומת v.

גרף אי פריק: גרף ללא צמתי הפרדה.

רכיב אי פריק: קבוצה מקסימלית של קשתות בה כל זוג קשתות נמצאות על מעגל פשוט.

עץ פורש: עץ אשר כולל את כל הצמתים בגרף.

עץ פורש מינימום: עץ אשר משקלו מינימלי מבין כל העצים הפורשים את הגרף

אלגוריתם "חמדן": אלגוריתם אשר מבצע בכל איטרציה את הדבר הנראה כרגע בתור הצעד הטוב ביותר (לא חושב לעתיד).

חתך בגרף: קבוצת קשתות המחברות בין 2 קבוצות של צמתים.

גרעין: הגרעין של G- S(G) הוא אוסף הקשתות ששייכות למסלול קצר ביותר מ-s ל-t.

שידוך: בגרף לא מכוון, תת קבוצה של קשתות כך שלאף זוג קשתות קצה משותף.

שידוך מקסימלי: תת הקבוצה המקסימלית של קשתות בגרף המהווה שידוך.

גודל שידוך: מספר הקשתות בשידוך

גרף דו צדדי: גרף לא מכוון אשר ניתן לחלק את צמתיו לשתי קבוצות זרות ומשלימות X, ו-Y כך שלכל קשת בגרף עוברת אך ורק בין X ל-Y ולא בתוך הקבוצות עצמן.

שידוך מלא: שידוך בגרף דו צדדי בו כל צומת מ-X משודך.

משפטים כלליים:

1. בהנתן גרף G(V,E), מספר הצמתים שדרגתם אי זוגית הוא זוגי
2. משפט אוילר: בגרף קשיר קיים מסלול אוילר אמ"מ יש בדיוק 2 צמתים שדרגתם אי זוגית, או שלכל הצמתים דרגה זוגית (ואז נקבל מעגל אוילר).
3. התנאים הבאים שקולים:
   - ❖ G הוא עץ ובו n צמתים
   - ❖ G חסר מעגלים אבל G+e מכיל מעגל לכל קשת e שחסרה ב-G
   - ❖ G חסר לולאות ולכל זוג צמתים קיים מסלול פשוט יחיד ב- G המחבר אותם.
   - ❖ G קשיר, אך אם נסיר קשת כלשהי הוא כבר לא יהיה קשיר
   - ❖ G חסר מעגלים ובו n-1 קשתות.
   - ❖ G קשיר ובו n-1 קשתות.

עצים מכוונים:

התנאים הבאים שקולים:

1. G עץ מכוון (יש שורש והתשתית עץ).
2. ל-G יש שורש וממנו מסלול יחיד לכל צומת.
3. ל-G יש שורש עם דרגת כניסה 0 ולכל צומת אחר יש דרגת כניסה 1.
4. ל-G יש שורש אבל השמטת כל קשת תפגע בתכונה זו.
5. גרף התשתית של G קשיר ויש בו צומת עם דרגת כניסה 0 ולכל צומת אחר דרגת כניסה 1.

שימוש ל DFS - מיון טופולוגי:

ממיין את הצמתים, לפי סדר, כך שאם ישנה קשת u-v אזי u יהיה לפני v במיון הטופולוגי. מריצים DFS על הגרף מי שנסוגים ממנו אחרון יהיה אחרון במיון הטופולוגי, כל פעם שניסוג ממישהו נוסיף אותו לתחילת הרשימה המקושרת. פועל רק ב-DAG (גרף מכוון ללא מעגלים). סיבוכיות כמו DFS: O(|V|+|E|)

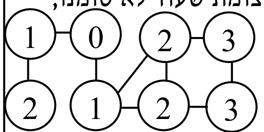
אלגוריתם BFS - Breadth First Search

אתחול: דרגת כל צומת אינסוף מלבד השורש שדרגתו 0.

עבור כל צומת, מסמן את כל השכנים של הצומת שעוד לא סומנו, בדרגה 1 יותר מהצומת הנוכחי.

יוצר את עץ המסלולים הקצרים ביותר.

סיבוכיות: O(|V|+|E|) עץ BFS אינו יחיד.

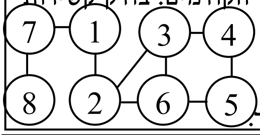


אלגוריתם DFS - Depth First Search

ממשיך לגלות צמתים חדשים עד אשר אין שכנים חדשים, ואז חוזר אחורה ומנסה לגלות שכנים חדשים בצמתים הקודמים. בודק קשירות של גרף ומחשב רכיבים אי פריקים.

סיבוכיות: O(|V|+|E|)

שורש s הוא צומת הפרדה אמ"מ יש לו לפחות 2 בנים בעץ DFS. עץ DFS אינו יחיד.



שימוש ל DFS - מציאת רכיבים בלתי פריקים:

לכל צומת נשמור k(v) (מספרו ב-DFS) וכן L(v) (מהי הצומת הגבוהה ביותר אליה ניתן להגיע בתת העץ וע"י קשת אחורית אחת). מריצים DFS, בכל צומת שמגלים נעדכן k(v) בעת הגילוי ובתחילה L(v)=k(v). אם נראה קשת אחורית נעדכן L(v) אם קטן יותר מהקיים. בעת הנסיגה לאב נעדכן אם יש צורך את L(v) של האב לזה של הצאצא, ואם לא עדכנו הצומת הינו צומת הפרדה (פרט לאם זה השורש). השורש יהיה צומת הפרדה רק אם בריצת DFS יהיו לו שני בנים ומעלה.

משפטים:

1. לכל גרף פשוט יש לפחות 2 צמתים בעלי אותה דרגה.
2. משפט Dirac: G גרף פשוט לא מכוון אם לכל זוג צמתים u,v בגרף מתקיים $d(u)+d(v)\geq|v|$
3. אם הקשת (a,b) איננה קשת בכל גרף DFS אזי a אב קדמון או צאצא של b בגרף הDFS.
4. למת בני הדודים: אם קיימת קשת (a,b) בגרף G, אז בכל עץ DFS a הוא צאצא של b או להפך (a ו- b אינם בני דודים בגרף)
5. u היא צומת הפרדה בין צומת v לשורש s אמ"מ $l(v)\geq k(u)$ כאשר v צאצא של u.
6. תנאי פונקציית זרימה חוקית:
   - א. הזרימה בקשת קטנה מקיבולה
   - ב. ליכסון סימטרי $f(u,v)=-f(v,u)$
   - ג. שימור: לכל צומת שאינה בור או מקור, זרימה נכנסת שווה לזרימה יוצאת.
7. משפט Min Cut - Max Flow: תהה F פונקצית זרימה חוקית הטענות הבאות שקולות:
   - א. f פונקציית זרימה מקסימלית
   - ב. אין מסלול שיפור מ-s ל- t ב Gf (הגרף השיורי)
   - ג. קיים חתך שמפריד בין s ל-t וקיבולו |f| (הזרימה המקסימלית) - חתך מינימלי
8. תהה (u,v) קשת בגרעין S(G), נוסיף לגרף את הקשת (v,u) אזי הגרעין לא ישתנה.
9. גרף הוא דו צדדי אמ"מ הוא לא מכיל מעגל בעל אורך אי זוגי.
10. משפט Hall: בגרף דו-צדדי קיים שידוך מלא אם לכל תת קבוצה ב-X יש מספר שכנים גדול או שווה למספר הצמתים בתת הקבוצה.

מסלולים קלים ביותר:

1. כל תת מסלול של מסלול קל ביותר הוא מסלול קל ביותר בעצמו.
2. משקל המסלול הקל ביותר מ S ל- V קטן או שווה למשקל המסלול הקל ביותר מ S ל- U + משקל הקשת U-V.
3. מסלול קל ביותר הוא בהכרח מסלול פשוט.

רלקסציה: פעולה של תיקון האומדן. רלקסציה על קשת U-V: אם $d(v)>d(u)+w(u,v)$ אזי $d(v)=d(u)+w(u,v)$ ואבא של v הוא u.

אלגוריתם Dijkstra לחישוב מסלולים קלים ביותר:
(מתאים לגרפים עם משקלי קשתות אי שליליים בלבד!)
1. אתחל משקל כל צומת לאינסוף מלבד השורש שמשקלו 0.
2. הכנס את כל הצמתים לתור עדיפויות על פי משקל המסלול הקצר ביותר אליהן עד כה.
3. הוצא את הצומת בעלת המשקל הנמוך ביותר.
4. בצע רילקסציה לכל שכניה.

סיבוכיות: $O(|V|^2)$ או $O(|V|\log|V| +|E|)$ עם שימוש בערימת פיבונצ'י

עצים פורשים מינימום

הערה כללית למציאת עפ"מ:
כל האלגוריתמים לעפ"מים פועלים על גרפים לא מכוונים.

אלגוריתם Prim למציאת עפ"מ:
נתחיל מ-s (קודקוד כלשהו) בכל איטרציה מוסיפים לעץ את הקשת הקלה ביותר שמגלה צומת חדש (אלגוריתם חמדן). (בעצם החתך בין העץ לבין שאר הצמתים).
סיבוכיות $O(|V|^2)$

אלגוריתם Boruvka למציאת עפ"מ:
מתחילים מיער (כל צומת מהווה עץ). בכל איטרציה עבור כל עץ בחר את הקשת הקלה ביותר בינו לבין כל העצים האחרים (קשת אחת בלבד - וכוון אותה). נאחד את כל העצים בסיום שלב זה, נעצור כאשר יהיה לנו עץ פורש.
סיבוכיות : לא ידוע.

אלגוריתם קוד האפמן:

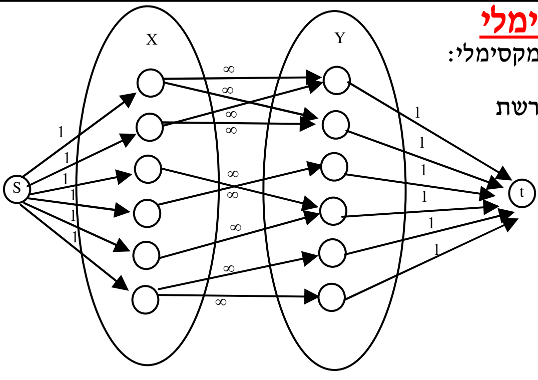
סדר את המשקלים בסדר לא יורד. אחד את השניים הנמוכים ביותר כשני עלים תחת אבא משותף. משקל העץ החדש הוא חיבור שני המשקלים. הכנס את העץ החדש חזרה למאגר במקום על פי משקלו החדש ובצע שוב עד קבלת איבר אחד בלבד. סיבוכיות זמן ריצה: $O(n\log(n))$

זרימה מקסימלית

אלגוריתם פורד פולקרסון למציאת זרימה מקסימלית:
אתחל זרימה בכל קשת ל-0. מצא מסלול כלשהו מ s ל- t. הרווה אותו. בנה רשת שיורית, ובצע שוב על הרשת השיורית, עד שאין יותר מסלולי שיפור.
זהו אלגוריתם בסיבוכיות פסואדו-פולינומית כלומר תלוי בערך הזרימה המקסימלית. וריאציית אדמונדס-קארפ מחליפה את צעד "מצא מסלול כלשהו" בהרץ BFS וקח מסלול קצר ביותר.
משפט: בכל איטרציה של אדמונדס קארפ, או שהמסלול הקצר ביותר מתארך, או שהגרעין קטן ממש.
סיבוכיות אדמונדס קארפ: $O(VE^2)$

מציאת שידוך מקסימלי

אלגוריתם למציאת שידוך מקסימלי:
1. בנה רשת זרימה כנ"ל:
2. מצא זרימת מקסימום ברשת זאת.
3. השידוך המקסימלי הוא כל קשת בין X ל-Y שיש עליה זרימה.



אלגוריתם Bellman Ford לחישוב מסלולים קלים ביותר:
(מתאים לגרפים עם משקלי קשתות ממשיים. מסוגל לזהות מעגל שלילי)
האלגוריתם מבצע $|V|-1$ פעמים רלקסציות על כל הקשתות בסדר כלשהו.
אם באיטרציה ה-V מתקבל שיפור כלשהו, מצאנו מעגל שלילי.
סיבוכיות: $O(|V|*|E|)$
אם קיים מסלול קל ביותר מ-s ל- v בעל k קשתות, הוא יתגלה **לכל היותר** באיטרציה ה- k.

אלגוריתם Floyd-Warshall לחישוב מסלולים קלים ביותר בין כל זוגות הצמתים:
אתחול - מטריצת שכנויות (משקל הקשת u-v או 0 בתא u-u)
נבצע n איטרציות כאשר בכל איטרציה i ננסה להוסיף את i כצומת ביניים במסלול בין כל שני צמתים. אם המסלול הקצר ביותר מכיל אינדקס מקסימלי k הוא יתגלה בוודאות באיטרציה ה-k לכל היותר. השורה ה-k והעמודה ה-k לא ישתנו במהלך האיטרציה ה-k. סיבוכיות: $O(|V|^3)$

אלגוריתם Johnson לחישוב מסלולים קלים ביותר בין כל זוג צמתים:
נוסיף צומת פיקטיבי s, מריצים בלמן-פורד לבדיקת מעגלים שליליים ואם אין, נשמור את התוויות, נסמנן $h(v)$, ונשנה את משקל הקשתות $e=(u-v)$ כך: $w'(e)=w(e)+h(u)-h(v)$. כעת נריץ דייקסטרא מכל צומת סה"כ $O(|V|^2\log|V|+|V||E|)$
ואם הגרף דליל כלומר $|E|=|V|\log|V|$ הסיבוכיות תהיה $O(|V|^2\log|V|)$. נחזיר את משקלי המסלולים למשקלים האמיתיים, נשים לב שמשקלו של מסלול p (מ-s ל- t) הוא $w'(p)=w(p)+h(s)-h(t)$. (בדייקסטרא שימוש בערימת פיבונצ'י).

אלגוריתם Kruskal למציאת עפ"מ:
נמיין את הקשתות ע"פ סדר לא יורד (עולה לא ממש). עבור כל קשת אם אינה סוגרת מעגל נוסיפה לעץ. (בדיקת "המנהיג", לכל קבוצה יש מנהיג, בתחילה כל אחד מנהיג של עצמו, אם מנהיגי הקבוצות שונים נהפוך את המנהיג של הקבוצה הגדולה למנהיג שתי הקבוצות). אורך רשימת המנהיגים קטנה או שווה ל $\log_2(v)$.
סיבוכיות : $O(|E|\log|E|)$

אלגוריתם גנרי למציאת עפ"מ:
הכלל הכחול: מצא חתך שאין בו קשת כחולה, צבע את הקשת הקלה ביותר בחתך בכחול.
הכלל האדום: מצא מעגל שאין בו קשת אדומה, צבע את הקשת הכבדה ביותר במעגל באדום.