חיפוש לרוחב – BFS:

BFS סורק את $G=(V,E)$ ומייצר עץ שבו כל הצמתים R שהם ברי הגעה מ- s. בנוסף, המרחק בקשתות על העץ מ- s לכל צומת ב- R הוא מינימאלי.

נותן פיתרון ל:

- מציאת מסלול קצר ביותר מהמקור לכל צומת.
- גלוי צמתים ברי הגעה מהמקור.

הרעיון:

החיפוש בוחן בכל איטרציה i "חזית" המורכבת מצמתים במרחק i מ- s. עבור $i \ge 0$ האלגוריתם יגלה את כל הצמתים במרחק i לפני כל הצמתים במרחק $(i+1)$ מ- s.

האלגוריתם:

קלט: גרף $G=(V,E)$ וצומת מקור s.

פלט: לכל $v \in V$, $d[v]$, המרחק מ- s ל- v.

for each $u \in V \setminus \{s\}$:

$d[u]=\infty,\quad p[u]=null$.

$Q=empty$

$enqueue(Q,s)$

while Q is not empty do:

$u=dequeue(Q)$

for each v in $adj(u)$ do:

$d[v]=d[u]+1$

if $p[v]=u$

$enqueue(Q,v)$

סיבוכיות: $O(V+E)$.

למה-1: לכל קשת (u,v) מתקיים: $dist(s,v) \le dist(s,u)$.

למה-2: בסיום האלגוריתם, הערך $d[v]$ מקיים $d[v] \ge dist(s,v)$ לכל צומת v.

למה-3: נניח שבמהלך ביצוע BFS התור מכיל את $\{v_1,\cdots,v_n\}$, אז מתקיים:

- $d[v_r] \le d[v_1]+1$.
- $d[v_i] \le d[v_{i+1}]$

כלומר: בתור יש צמתים משתי רמות לכל היותר, ויש מונוטוניות ברמות הצמתים שבתור.

מסקנה: נניח ש- v_i נכנס ל- Q לפני v_j, אז בזמן ש- v_j נכנס לתור מתקיים: $d[v_i] \le d[v_j]$.

משפט: הרצת BFS על גרף G (מכוון או לא מכוון) עם צומת מקור s, מגלה כל צומת ישיג מ- s ובסיום האלגוריתם: $d[v]=dist(s,v)$.

מיון טופולוגי:

מיון טופולוגי של G הוא סידור ליניארי של V כך שאם $uv \in E$, אז u לפני v בסידור.
אם בגרף יש מעגל מכוון אז אין לגרף מיון טופולוגי.
DAG (Directed Acyclic Graph)- גרף מכוון שאין בו מעגלים מכוונים.

אלגוריתם למיון טופולוגי של DAG:

קלט: גרף $G=(V,E)$ שהוא DAG.

פלט: מיון טופולוגי של הגרף.

1. חשב את קבוצת כל המקורות בגרף, נסמנה ב- S.
2. אתחל $l \leftarrow 1$.
3. כל עוד $V \neq \varnothing$:
   3.1 בחר $v \in S$
   3.2 קבע $L(v) \leftarrow l$
   3.3 קבע $l \leftarrow l+1$.
   3.4 הסר את v מהגרף, יחד עם הקשתות היוצאות ממנו.
   3.5 קבע $S \leftarrow S \setminus \{v\}$.
   3.6 הוסף ל- S את כל הצמתים מבין $\{u : vu \in E\}$ שהם מקורות.
4. החזר את L.

סיבוכיות: $O(V+E)$.

טענה-1: בכל DAG $G=(V,E)$ כך ש $V \neq \varnothing$, קיים מקור.

סיבוכיות ואורך הקלט:

לצורך ייצוג מספר בגודל n נדרשים: $m=\log_2 n$ ביטים ולכן אורך הקלט הוא: $\Omega(\log n)$. אם יש לנו מספר n בקלט, הסיבוכיות צריכה להיות פולינומיאלית ב- $\log n$. אם זה לא המצב והסיבוכיות היא פולינומיאלית דווקא ב- n, נאמר שהאלגוריתם- **פסאודו פולינומיאלי**.

אם נתון לנו גרף $G=(V,E)$ אז הסיבוכיות צריכה להיות פולינומיאלית ב- $|V|,|E|$.

גרף דו-צדדי:

בהינתן גרף לא מכוון $G=(V,E)$, האלגוריתם מכריע האם G הוא גרף דו-צדדי.

טענה-1: יהא $G=(V,E)$ גרף לא מכוון, ויהא $s\in V$. G הוא גרף דו-צדדי אם"ם $uv\in E \Rightarrow d_s(u)\not\equiv_2 d_s(v)$ (שקילות מודולו 2).

טענה-2: יהא $G=(V,E)$ גרף דו-צדדי עם צדדים L,R. יהא p מסלול בין u ו- v. אזי אורך המסלול p הוא זוגי אם"ם u,v נמצאים באותו צד.

האלגוריתם:

1. הרץ BFS מצומת שרירותי $s\in V$.
2. החזר "G דו צדדי" אם"ם לכל קשת $uv\in E$ מתקיים $d_s(u)\not\equiv_2 d_s(v)$.

סיבוכיות: $O(V+E)$.

אפיון נוסף: גרף הוא דו-צדדי אם"ם הוא אינו מכיל מעגלים באורך אי-זוגי.

מסלולים זוגיים קצרים ביותר:

רדוקציה:

שכפול צמתי הגרף לקבוצה-1 וקבוצה-2 וחיבור קשתות בין צמתים מעותקים בקבוצות שונות, אשר בגרף המקורי הייתה קשת שחיברה בין הצמתים המתאימים. זהו גרף דו-צדדי.

טענה: יהא $G=(V,E)$ גרף לא מכוון ויהא $s\in V$. אז $d_s^{even}(v)=d_{s^1}(v^1)$.

האלגוריתם:

קלט: גרף $G=(V,E)$ וצומת $s\in V$.

פלט: לכל צומת $v\in V$, אורך המסלול הקצר ביותר מ- s ל- v מבין המסלולים שאורכם זוגי, או ∞ אם לא קיים מסלול כזה.

1. בנה את הגרף $G'=(V^1\cup V^2,E')$ (לכל קשת $uv\in E$ נגדיר שתי קשתות $u^1v^2,u^2v^1\in E'$.
2. הרץ BFS על G' החל מהצומת s^1.
3. לכל צומת $v\in V$ החזר: $d_{s^1}(v^1)$.

סיבוכיות: $O(V+E)$.

חיפוש לעומק – DFS:

נותן פיתרון ל:

- מציאת מעגלים בגרף.
- מציאת שורשים בגרף.
- כיוון קשתות הגרף.

הרעיון:

האלגוריתם ינסה להתקדם לעומק, כאשר נבקר בצומת v, אם ישנה קשת (v,u) לצומת u שעוד לא נתגלה- נחצה את הקשת ונמשיך את החיפוש מ- u.

האלגוריתם:

קלט: גרף $G=(V,E)$ וצומת s ב- V.

פלט: לכל $v \in V$ נקבל $d[v]$ - זמן הגילוי של v ו- $p[v]$, הצומת שגרם ל- v להתגלות.

סימון: $color[u]$ צבע לבן או אפור. $f[u]$ - זמן היציאה האחרון מ- u.

```
for each u ∈ V :
        color[u] = white
        p[u] = null
i = 0
for each u ∈ V :
        if color(u) = white then:
                DFS_VISIT(u)
DFS_VISIT(u):
        color(u) = gray
        i++
        d[u] = i
        for each v in adj(u):
                if color(v) = white then:
                        p[v] = u
                        DFS_VISIT(v).
        color(u) = black.
        i++
        f[u] = i
```

סיבוכיות: $O(V+E)$.

הערות:

– בהרצות שונות ייתכנו פלטים שונים לאלגוריתם.
– DFS ימצא גם צמתים שאינם ישיגים מ- s.
– DFS אינו בהכרח מוצא מרחקים קצרים ביותר.

בסיום האלגוריתם נקבל גרף G_π, זהו תת גרף של G שבו לכל צומת v שעבורו $p[v] \neq null$, תופיע הקשת מהאב-$p[v]$ אל v כפי שנמצא ע"י האלגוריתם.

סיווג קשתות:

קשתות עץ: מופיעות ב- G_π.

קשתות אחוריות: קשת (u,v) שמחברת את u לאב קדמון של u בעץ DFS (לולאה עצמית תחשב כקשת אחורית).

קשתות קדמיות: קשתות שאינן ב- G_π. כל קשת (u,v) מ- u לצאצא של u בעץ DFS.

קשתות חוצות: כל הקשתות האחרות ב- G בין צמתים באותו עץ DFS ללא יחס אב קדמון-צאצא, או בין עצי DFS שונים.

תכונות של DFS:

– הגרף G_π הוא יער, היות והמבנה של עצי החיפוש משקף את הקריאות לDFS_VISIT.
– הצומת v הוא צאצא של u בעץ DFS אם v נתגלה כאשר u היה אפור ולפני שקבענו ערך ל $f[u]$.
– תכונת הסוגריים: נייצג את הגילוי של צומת v ע"י סוגר שמאלי $(v$ ואת סיום הטיפול ב- v ע"י סוגר ימני $v)$. אז ההיסטוריה של גילוי וסיום הטיפול בצמתים מגדירה ביטוי שבו הסוגריים מקוננים היטב. למשל: $(u(v(y(xx)y)v)u)(w(zz)w)$.

משפט הסוגריים: בכל חיפוש לעומק בגרף מכוון או לא מכוון G, לכל שני צמתים u,v מתקיים בדיוק אחד מ-3 התנאים:

1. האינטרוולים $[d[u], f[u]]$ ו- $[d[v], f[v]]$ זרים לחלוטין ואז אין קשר של אב קדמון/צאצא בין הצמתים u,v.
2. האינטרוול $[d[u], f[u]]$ מוכל ממש ב $[d[v], f[v]]$, ואז u צאצא של v בעץ DFS.
3. האינטרוול $[d[v], f[v]]$ מוכל ממש ב- $[d[u], f[u]]$ ואז u אב קדמון של v בעץ DFS.\

מסקנה: צומת v הוא צאצא של צומת u ביער DFS עבור גרף מכוון/לא מכוון G אם"ם $d[u] < d[v] < f[v] < f[u]$.

משפט המסלול הלבן: ביער DFS מכוון/לא מכוון, צומת v הוא צאצא של u אם"ם בזמן $d[u]$, הזמן בו u התגלה, ניתן להגיע מ- u ל- v על מסלול המורכב כולו מצמתים לבנים.

תכונות מהתרגול:

- בגרפים מכוונים יש רק קשתות עץ או קשתות אחוריות.
- הרצת DFS תסווג קשת כאחורית אם"ם יש מעגל בגרף.
- בהרצת DFS על גרף מכוון $G=(V,E)$, נסמן את הקשתות שסווגו כאחוריות ב- $F \subseteq E$. אז הגרפים (V,F) ו- $(V,E \setminus F)$ הם DAG.
- הסדר ההפוך לזה המושרה ע"י זמני הנסיגה מהצמתים הוא מיון טופולוגי.
- בכל הרצת DFS, עץ ה-DFS האחרון מכיל את R_G - קבוצת שורשי הגרף.

רכיבים קשירים היטב:

רכיב קשיר היטב: בגרף מכוון $G=(V,E)$ זו קבוצה מקסימאלית של צמתים C ב- V כך שלכל זוג צמתים u,v ב- C יש מסלול מכוון מ- v ל- u ומ- u ל- v. (ייתכן מצב בו צומת בודד הוא הכי קשיר היטב).

הגרף ההפכי: עבור גרף מכוון $G=(V,E)$ הוא $G^T=(V,E^T)$ כך ש $E^T=\{(u,v)\,|\,(v,u)\in E\}$

דהיינו, הופכים את כיווני הקשתות ב- G. ב- G וב- G^T יש אותם רכיבים קשירים היטב.

ניתן ליצור את G^T ב- $O(V+E)$.

האלגוריתם:

1. מצא בעזרת DFS את ערכי $f[v]$ לכל צומת.
2. הפוך את כיווני הקשתות וצור את הגרף G^T.
3. הפעל DFS על הגרף ההפוך אבל בחר את הצמתים בלולאה הראשית לפי סדר יורד של $f[v]$.
4. כל עץ DFS שמתקבל הוא רכיב קשיר היטב.

סיבוכיות: $O(V+E)$.

למה-1: יהיו C,C' שני רק"ה שונים בגרף מכוון- G. ויהיו u,v ב- C ו u',v' ב- C'. נניח שיש מסלול מכוון מ- u ל- u' ב- G, אז לא ייתכן מסלול מכוון מ v' ל- v.

מסקנה: גרף הרכיבים הוא DAG.

הרחבת הגדרות "זמן גילוי" ו"זמן סיום" לקבוצת צמתים, $U\subset V$:

$$d(U)=\min_{u\in U}\{d(u)\}$$
$$f(U)=\max_{u\in U}\{f(u)\}$$

למה-2: יהיו C,C' שני רק"ה בגרף מכוון $G=(V,E)$. נניח שיש קשת (u,v) כאשר $u\in C, v\in C'$, אז $f(C)>f(C')$.

מסקנה: יהיו C,C' רק"ה שונים בגרף מכוון $G=(V,E)$. נניח שיש קשת $(u,v)\in E^T$ כאשר $u\in C, v\in C'$, אז: $f(C')>f(C)$.

גרף הרכיבים הקשירים היטב:

גרף G^{SCC} (Strongly Connected Components) הוא גרף אשר צמתיו מייצגים רכיבים קשירים היטב ב- G וקשת מחברת בין שני צמתים u,v אם"ם קיים צומת u' ברכיב הקשיר היטב המיוצג ע"י u וקיים צומת v' ברכיב הקשיר היטב המיוצג ע"י v כך ש- $u'v'$ קשת ב- G.

מציאת שורשי הגרף:

טענה-1: בהרצת DFS על גרף מכוון החל מצומת u, מתקבל עץ DFS יחיד ביער ה-DFS אם"ם u הוא שורש של הגרף.

אלגוריתם למציאת שורש אחד:

קלט: גרף מכוון $G=(V,E)$.

פלט: שורש של הגרף או הודעה כי $R_G \neq \varnothing$.

1. הרץ DFS על G.
2. נסמן ב- v את שורש עץ ה-DFS האחרון.
3. הרץ DFS על G החל מ- v.
4. אם התקבל עץ DFS אחד החזר את v, אחרת החזר כי $R_G=\varnothing$.

סיבוכיות: $O(V+E)$.

טענה-2: אם $R_G \neq \varnothing$ אז R_G הוא רכיב קשיר היטב.

טענה-3: אם $R_G \neq \varnothing$ אז הצומת הראשון במיון הטופולוגי של G^{SCC} הוא R_G.

רכיבים קשירים היטב ועצי DFS:

טענה-4: יהא $G=(V,E)$ גרף מכוון ויהא U רק"ה של G. אם p הוא מסלול מ- u ל- v כאשר $u,v\in U$ אז p מוכל בשלמותו ב- U.

משפט: יהא $G=(V,E)$ גרף מכוון, ויהא U רק"ה של G. אז לכל ריצת DFS, U מוכל בשלמותו בעץ DFS.

גרפים קשירים למחצה:

קשיר למחצה: גרף מכוון $G=(V,E)$ שעבור כל זוג צמתים u,v קיים מסלול מכוון מ- u ל- v או מסלול מכוון מ- v ל- u (או שניהם).

האלגוריתם:

קלט: גרף מכוון $G=(V,E)$.

פלט: הכרעה בדבר האם G קשיר למחצה.

1. חשב את G^{SCC}.
2. הרץ מיון טופולוגי על G^{SCC}.
3. הכרז "הגרף קשיר למחצה" אם"ם יש קשת בין כל שני צמתים עוקבים במיון.

עצים פורשים מינימאליים:

בהינתן גרף קשיר לא מכוון $G=(V,E)$ שבו לכל קשת (u,v) יש משקל $w(u,v)$, יש למצוא עץ פורש של הגרף שסה"כ משקל הקשתות בו הוא מינימאלי.

הרעיון:

נתקדם ע"י צביעת קשתות תוך הוספת קשתות קלות לעץ והשמטת קשתות כבדות. האלגוריתם יקיים בכל של את **שמורת הצבע**- קיים עפ"מ שמכיל את כל הקשתות הכחולות ואף אחת מהקשתות האדומות.

חתך (cut): חתך בגרף $G=(V,E)$ הוא חלוקה של V לשתי קבוצות זרות X ו- $V\setminus X$. נאמר שקשת חוצה את החתך אם קצה אחד שלה ב- X והשני ב- $V\setminus X$.

האלגוריתם הגנרי:

הכלל האדום: בחר מעגל חסר קשתות אדומות וצבע באדום את הקשת הלא צבועה הכבדה ביותר.

הכלל הכחול: בחר מעגל חסר קשתות כחולות וצבע בכחול את הקשת הלא צבועה הקלה ביותר.

האלגוריתם החמדן: הפעל את הכלל האדום והכחול (לא דטרמיניסטית) עד שכל קשתות הגרף צבועות. הקשתות הכחולות מהוות עפ"מ.

משפט-1: האלגוריתם הגנרי צבוע את כל הקשתות של גרף קשיר G ומקיים את שמורת הצבע.

האלגוריתם של Prim:

1. הגדר עץ T והכנס לתוכו צומת כלשהו בגרף.
2. כל עוד יש בגרף קשתות לא צבועות:
   - 2.1 בחר קשת קלה ביותר בחתך בין T לשאר הגרף וצבע אותה בכחול.
   - 2.2 הכנס את הצומת אליו הייתה הקשת אל T.
   - 2.3 צבע באדום את כל הקשתות האחרות בחתך שמובילות אל הצומת.
3. כל הקשתות הכחולות מגדירות עפ"מ.

סיבוכיות במימוש בעזרת מערך: $O(V^2)$.

סיבוכיות במימוש בעזרת ערימה: $O(E\log V)$.

האלגוריתם של Kruskal:

1. מיין את הקשתות לפי משקל בסדר לא יורד ואתחל את T להיות ריק.
2. עבור על הרשימה הממוינת ולכל קשת e בצע:
   - 2.1 אם e סוגרת מעגל בעץ הכחול, צבע את e באדום
   - 2.2 אחרת, צבע את e בכחול ובצע: $T\leftarrow T\cup\{e\}$.
3. החזר את אוסף קשתות T כעפ"מ.

סיבוכיות: $O(E\log E)=O(E\log V)$ (כי $\log E\le\log V^2=2\log V$).

Kruskal מגדל את העפ"מ כיער ההולך ומתאחד, Prim מתחיל מעץ קטן ובכל שלב מוסיף לו צומת.

האלגוריתמים הללו הם "שלמים"- כל עפ"מ יכול להימצא על ידם.

עפ"מ צהוב ביותר:

כל קשת בגרף צבועה בצהוב או בשחור. נרצה לתת לקשתות צהובות עדיפות על-פני קשתות שחורות מאותו משקל. לשם כך נגדיר פונקציית משקל חדשה:

$$w'(e)=\begin{cases} w(e)-1/n^2 & e \text{ is yellow} \\ w(e) & \text{otherwise} \end{cases}$$

טענה-1: T הוא עפ"מ צהוב ביותר לפי פונקציית המשקל w אם"ם T הוא עפ"מ לפי פונקציית המשקל w'.

הערה: המונוטוניות של משקלי העצים נשמרת ללא קשר לצבע הקשתות.

האלגוריתם:

קלט: גרף לא מכוון וקשיר $G=(V,E)$ ופונקציית משקל $w:E\to N$.

פלט: עפ"מ צהוב ביותר של G.

1. חשב את פונקציית המשקל w'.
2. מצא עפ"מ T לפי פונקציית המשקל w' והחזר את T כפלט.

סיבוכיות: $O(E\log V)$.

הערה: בבעיית עפ"מ, כל שמשנה הוא סדר המשקלות ולא ערך המשקלות עצמם. ולכן אם שתי פונקציות משקל שומרות על אותו יחס סדר בין המשקלות, כל עפ"מ לפי פונקציה ראשונה יהיה עפ"מ אם"ם הוא עפ"מ לפי הפונקציה השנייה.

מספר קשתות ממשקל זהה בעפ"מ:

טענה-2: יהא $G=(V,E)$ גרף לא מכוון וקשיר עם פונקציית משקל $w:E\to R$. אז לכל משקל נתון, כל עפ"מ של G מכיל את אותו מספר של קשתות ממשקל זהה.

האם קיים עפ"מ המכיל קשת מסוימת.

האלגוריתם הלא יעיל:

קלט: גרף $G=(V,E)$ לא מכוון וקשיר עם פונקציית משקל $w:E\to R$, וקשת $e\in E$.

פלט: הכרעה בדבר האם קיים עפ"מ של G המכיל את e.

1. צבע את e בצהוב ואת שאר הקשתות בשחור.
2. הרץ את האלגוריתם למציאת עפ"מ צהוב ביותר. נסמן את העץ המתקבל ב- T.
3. החזר "כן" אם"ם e היא קשת ב- T.

סיבוכיות: $O(E\log V)$.

טענה-1: קשת e נמצאת באיזשהו עפ"מ של G אם"ם כל מעגל המכיל את e, מכיל קשת $e'\neq e$ כך ש- $w(e')\geq w(e)$.

האלגוריתם היעיל:

1. בנה את הגרף $G'=(V,E')$ כך ש- $E'=\{e'\in E: w(e')<w(e)\}$
2. החזר "כן" אם"ם לא קיים מסלול המחבר בין שני צמתים u,v ב- G'.

סיבוכיות: $O(V+E)$.

האם קשת מסוימת מופיעה בכל העפ"מים:

האלגוריתם הלא יעיל:

קלט: גרף $G=(V,E)$ לא מכוון וקשיר עם פונקצית משקל $w:E\to R$, וקשת $e\in E$.

פלט: הכרעה בדבר האם כל עפ"מ של G מכיל את הקשת e.

1. צבע את e בשחור ואת שאר הקשתות בצהוב.
2. הרץ את האלגוריתם למציאת עפ"מ צהוב ביותר. נסמן את העץ המתקבל ב- T.
3. החזר "כן" אם"ם e היא קשת ב- T.

סיבוכיות: $O(E\log V)$.

אלגוריתם לא יעיל נוסף:

1. חשב את משקלו של עפ"מ של G.
2. הסר את e וחשב את משקלו של עפ"מ בגרף החדש (אם בכלל קיים).
3. אם לא קיים עפ"מ בגרף החדש, או שמשקלו של העפ"מ בגרף החדש גדול ממשקלו של העפ"מ בגרף המקורי, החזר: "כן". אחרת, החזר: "לא".

סיבוכיות: $O(E\log V)$.

טענה-2: קשת e נמצאת בכל עפ"מ של G אם"ם כל מעגל המכיל את e מכיל קשת e' כך ש- $w(e')>w(e)$.

האלגוריתם היעיל:

1. בנה את הגרף $G'=(V,E')$ כך ש- $E'=\{e'\in E: w(e')\le w(e)\}\setminus\{e\}$
2. החזר "כן" אם"ם לא קיים מסלול המחבר בין שני צמתים u,v ב- G'.

סיבוכיות: $O(V+E)$.

מסלולים קלים ביותר:

מסלול: בהינתן גרף מכוון/ לא מכוון $G=(V,E)$ שבו לכל קשת (u,v) משקל $w(u,v)$, המשקל של מסלול $p=v_0v_1\cdots v_k$ הוא סכום המשקלות על הקשתות $w(p)=\sum_{i=1}^{k}w(v_{i-1},v_i)$.

מסלול קל ביותר: מסלול קצר ביותר מ- u ל- v הוא כל מסלול p שמקיים: $w(p)=dist(u,v)$.

מסלולים קצרים ממקור יחיד:

טענה-1: יהי p מסלול קל ביותר מ- s ל- u אזי כל תת-מסלול של p, גם הוא קל ביותר.

טענה-2 (אי-שוויון המשולש): לכל קשת $e=(u\to v)$ מתקיים: $\delta(s,v)\le\delta(s,u)+w(u\to v)$.

הרעיון: נעדכן כל הזמן הפרות של אי-שוויון המשולש

האלגוריתם הגנרי:

1. $d(s)\leftarrow 0$, ולכל v, $d(v)\leftarrow\infty$.
2. כל עוד יש קשת (u,v) עבורה סימוני d מפרים את אי-שוויון המשולש בצע שיפור מקומי:

$$d(v)\leftarrow d(u)+w(u,v)$$

3. כשאין קשת עליה ניתן לבצע שיפור מקומי, אז לכל צומת מתקיים: $d(v)=\delta(v)$.

האלגוריתם של דייקסטרה:

קלט: גרף ממושקל מכוון ללא קשתות שליליות וצומת s.

פלט: לכל צומת v - $d(v)=dist(s,v)$.

1. לכל $v\in V$ אתחל $d(v)\leftarrow\infty$ ואתחל $d(s)\leftarrow 0$.
2. ואתחל $Q\leftarrow V$ (Q - קבוצת הצמתים עבורם $d(v)\ne\delta(v)$).
3. כל עוד $Q\ne\varnothing$ בצע:

   3.1 מצא ב- Q צומת u בעל סימון $d(u)$ מינימאלי.

   3.2 הוצא את u מ- Q ולכל קשת (u,v) בצע שיפור (u,v).

סיבוכיות: $O(E\log V)$.

טענה-1: יהי $\bar{d}(u)$ הערך $d(u)$ כאשר u הוצא מ- Q. אם v הוצא מ- Q באיטרציה העוקבת אז: $\bar{d}(u)\le\bar{d}(v)$.

מסקנה-1: הטענה נכונה גם לאיטרציות שאינן עוקבות.

מסקנה-2: הסימון של צומת v $d(v)$, אינו משתנה אחרי הוצאת v מ- Q.

טענה-2: בסיום ריצת האלגוריתם לכל קשת (u,v) מתקיים: $d(v)\le d(u)+w(u,v)$, כלומר- לא ניתן לשפר מקומית.

האלגוריתם של בלמן-פורד:

קלט: גרף מכוון $G=(V,E)$ ממושקל, ללא מעגלים שליליים וצומת s.

פלט: לכל צומת v - $d(v)=dist(s,v)$.

1. לכל $v\in V$ אתחל: $d(v)\leftarrow\infty$. ואתחל $d(s)\leftarrow 0$.
2. בצע $|V|-1$ פעמים:

   2.1 לכל קשת (u,v) בצע שיפור מקומי אם צריך.

סיבוכיות: $O(VE)$.

טענה: לכל צומת v, אם קיים מסלול קל ביותר מ- s ל- v שמכיל k קשתות, אז לאחר k איטרציות: $d(v)=dist(s,v)$.

מציאת מסלולים קלים ביותר עם קשת אחת שלילית:

האלגוריתם:

קלט: גרף מכוון $G=(V,E)$ עם משקלות אי-שליליים על הקשתות, פרט לקשת אחת $e=(a,b)$ שמשקלה שלילי. וכן, הגרף חסר מעגלים שליליים. ונתון צומת $s\in V$.

פלט: לכל צומת $v\in V$, משקל המסלול הקל ביותר מ- s ל- v.

1. הסר את e מ- G.
2. בגרף המתקבל חשב את $d_s(v)$ ואת $d_b(v)$ לכל $v\in V$, בעזרת שתי הרצות דייקסטרה.
3. לכל צומת $v\in V$ החזר את הערך: $\min\{d_s(a)+w(e)+d_b(v),d_s(v)\}$.

סיבוכיות: $O(V+E\log V)$.

All-Pairs Shortest Paths:

טענה-1: תהא $h:V\to R$ פונקציית משקל המתאימה לכל צומת מספר ממשי. נגדיר $w_h:E\to R$ באופן הבא: לכל קשת $uv\in E$ נגדיר: $w_h(uv)=h(u)+w(uv)-h(v)$. אזי:

1. מסלול p מ- u ל- v הוא קל ביותר לפי w אם"ם הוא קל ביותר לפי w_h.
2. אין מעגלים שליליים לפי w_h.

האלגוריתם:

קלט: גרף מכוון $G=(V,E)$ עם פונקציית משקל על הקשתות $w:E\to R$ ללא מעגלים שליליים.

פלט: משקל מסלול קל ביותר מכל צומת לכל צומת.

1. בנה גרף $G'=(V',E')$ כך ש- $V'=V\cup\{s\}$ ו- $E'=E\cup\{sv:v\in V\}$.
2. חשב את $\delta_s(v)$ לכל v בעזרת האלגוריתם בלמן-פורד.
3. חשב את פונקציית המשקל $w_{\delta_s}=w(uv)+\delta_s(u)-\delta_s(v)\ge 0$.

4. מכל צומת $u \in V$ הרץ דייקסטרה לחישוב משקל מסלולים קלים ביותר מ- u לכל הצמתים, ביחס לפונקציית המשקל w_{δ_s}. נסמן את הערך המתקבל ב- $d'(u,v)$.

5. עבור כל זוג צמתים $u,v \in V$ החזר כפלט: $d'(u,v)+\delta_s(v)-\delta_s(u)$.

סיבוכיות: $O\left(V^2+EV\log V\right)$.

מציאת מעגל שלילי בגרף:

האלגוריתם:

קלט: מטריצת שערי חליפין A של n סוגי מטבעות.

פלט: קיומה של סדרת החלפת מטבעות המבטיחה לנו רווח.

1. בנה את הגרף $G=(V,E)$ על n צמתים. כל צומת ייצג מטבע. בין כל זוג צמתים i,j תהיה קשת עם משקל $w_{i,j}$. ונגדיר $w_{i,j}=-\log\left(A_{i,j}\right)$.

2. הרץ את בלמן-פורד מצומת שרירותי $s \in V$ להכרעה בדבר קיומו של מעגל שלילי בגרף, והחזר כמוהו.

סיבוכיות: $O(VE)=O\left(n^3\right)$.

הערה: מכיוון שכל צמתי הגרף הם שורשים, אם קיים מעגל שלילי אז הוא ישיג מכל צומת ולכן ניתן לבחור בשלב (2) צומת שרירותי.

אלגוריתמים חמדניים:

בגישה חמדנית מבצעים בכל שלב את הצעד הטוב ביותר כרגע.

קבוצה ב"ת גדולה ביותר בגרף אינטרוולים:

נתונות n משימות, לכל משימה i נתון זמן התחלה s_i וזמן סיום f_i. יש מכונה בודדת המסוגלת לבצע בכל רגע נתון משימה אחת לכל היותר.

גרף אינטרוולים: גרף לא מכוון כך ש:

1. אוסף הצמתים הם אינטרוולים.
2. יש קשת בין שני אינטרוולים אמ:ם הם נחתכים.

מחפשים תת-קבוצה של צמתים שבין כל שניים אין קשת- "זרים בזוגות".

קבוצה בלתי תלויה: קבוצה $X \subseteq V$ שבה לכל $u,v \in X$ אין קשת $(u,v) \in E$.

האלגוריתם:

קלט: אוסף משימות $S = \{a_1, \cdots, a_n\}$. לכל משימה זמן התחלה s_i וזמן סיום f_i.

פלט: תת-קבוצה מקסימאלית $A \subseteq S$ של משימות (או אינטרוולים) הזרים בזוגות.

1. אתחל $\varnothing \leftarrow A$ ומיין את המשימות לפי זמני סיום לא יורדים (ונוכל להניח מעכשיו ש- $f_1 \le f_2 \le \cdots \le f_n$).
2. בנוסף אתחל $0 \leftarrow i$ ו- $-\infty \leftarrow f_0$. (i - האינדקס האחרון שהוכנס ו- f_i זו נקודת הסיום שלו).
3. עבור $m = 1$ עד n בצע:

   3.1 אם a_m זר לכל המשימות ב- A, אז $A \leftarrow A \cup \{a_m\}$

סיבוכיות: $O(n \log n)$.

הערה: הרעיון הוא שאם המשימה שהסתיימה אחרונה ב- A (המזוהה עם האינדקס i) לא נחתכת עם a_m אז אף-אחת לא נחתכת איתו.

טענה-1: בסיום האיטרציה ה- k, קיים פיתרון אופטימאלי X^* כך שלכל $1 \le j \le k$, $I_j \in X^* \leftrightarrow I_j \in X$.

עצי הופמן וקודים פרפיקסים:

נתונים n תווים, רוצים לקודד (להחליף) כל אחד מהם במחרוזת בינארית שונה. המטרה היא למצוא קידוד עבור טקסט המביא למינימום את אורך הטקסט המקודד.

מילה: רצף של ביטים.

קוד: אוסף לא ריק של מילים.

קוד חד-פענח: בהינתן סדרה כלשהי של אותיות, אחרי שנקודד את הסדרה נוכל גם לשחזר אותה. שתי סדרות שונות לא יקודדו לאותה סדרת ביטים.

קוד פרפיקסי: קוד שבו אף מילה היא לא פרפיקס של מילה אחרת. קוד פרפיקסי הוא חד-פענח.

התאמה בין קודים פרפיקסים לעצים בינאריים:

לכל קוד פרפיקסי מתאים עץ בינארי. עלי העץ הם מילות הקוד. יהי T עץ בינארי כזה ותהי x אות שקודדנו. המחרוזת שתייצג את התו x תופיע כמסלול משורש העץ לעלה המתאים.

נסמן ב- $d_T(x)$ את עומק העלה ששווה גם לאורך מילת הקוד שמתאימה ל- x. אם לכל אות x יש תדירות $f(x)$, מספר הביטים הדרוש לקידוד הקובץ הוא: $\boxed{B(T) \triangleq \sum_x f(x) d_T(x)}$. נאמר כי T הוא אופטימאלי (עבור הקוד והתדירות הנתונים) אם הסכום הנ"ל הוא המינימאלי האפשרי.

הערה: ניתן להראות כי קוד אופטימאלי ייוצג ע"י עץ בינארי מלא (לכל צומת פנימי שני בנים). מספר העלים: n, מספר הצמתים הפנימיים: $n-1$.

למה-1: קיים עץ אופטימאלי בו שתי מילים x, y בעלות תדירויות מינימאליות מתאימות לזוג עלים אחים בעלי עומק מקסימאלי.

הרעיון: לעבור מבעיה עם n מילים- $C=\{(w_1,f_1),\cdots,(x,f(x)),(y,f(y))\cdots\}$ לבעיה עם $n-1$ מילים $C'=\{(w_1,f_1),\cdots,(z,f(z))\cdots\}$ כאשר: $f(z)=f(x)+f(y)$. ולאחר שנפתור בעיה זו עבור $n-1$ מילים, נחליף את z ונתייחס אליה כאל צומת פנימית עם בנים x, y.

למה-2: יהי C' הקוד המתקבל מ- C ע"י החלפת זוג אותיות בעלות תדירויות מינימאליות באות חדשה z כך ש $f(z)=f(x)+f(y)$, ויהי T' עץ אופטימאלי עבור C', אז העץ T המתקבל מ- T' ע"י הפיכת העלה המתאים ל- z לצומת פנימי ולו שני בנים x, y, הוא עץ אופטימאלי עבור C.

אלגוריתם הופמן רקורסיבי לקוד פרפיקסי אופטימאלי:

Recursive_Huffman(Q):

קלט: ערימה Q.

פלט: עץ הופמן של איברי Q.

1. אם $|Q|=2$, קודד אות אחת ב-'0' ואת השנייה ב-'1'.
2. אחרת,

   2.1 הוצא מ- Q 2 איברים x, y בעלי תדירות מינימאלית, והכנס ל- Q איבר חדש z שתדירותו $f(z)=f(x)+f(y)$.

   2.2 קרא ל- Recursive_Huffman(Q) שמחזיר את T'.

   הוסף לעלה המתאים ל- z את x ו- y כבנים ב- T', והחזר את העץ שהתקבל- T.

סיבוכיות: $O(n \log n)$.

קבוצה ב"ת גדולה ביותר ממושקלת עם אינטרוולים:

נתון אוסף משימות כאשר משימה מאופיינת כמו קודם בזמני התחלה וסיום אבל גם במשקל w_i. יש למצוא תת-קבוצה ב"ת של אינטרוולים שמשקלה הוא המקסימאלי האפשרי.

$opt(i)$ - המחיר של פיתרון אופטימאלי כלשהו עבור המשימות $\{a_1,\cdots,a_i\}$.

$pred(i)$ - האינדקס של האינטרוול שהסתיים מאוחר ביותר מבין האינטרוולים a_i שמסתיימים עד s_i, או 0 אם אין כאלה.

נגדיר $opt(0)=0$.

טענה: לכל $1\le i\le n$ מתקיים: $opt(i)=\max\begin{cases}opt(i-1)\\ opt(pred(i))+w_i\end{cases}$.

האלגוריתם:

1. אתחל $opt(0)\leftarrow 0$, $A(0)\leftarrow\varnothing$ ומיין את המשימות בסדר לא יורד לפי זמני הסיום.
2. לכל $1\le i\le n$ מצא את $pred(i)$.
3. לכל $1\le i\le n$ בצע:

   3.1 אם: $opt(i-1)<opt(pred(i))+w_i$ אזי:

   3.1.1 $A(i)\leftarrow A(pred(i))\cup\{a_i\}$

   3.1.2 $opt(i)\leftarrow opt(pred(i))+w_i$.

   3.2 אחרת:

   3.2.1 $A(i)\leftarrow A(i-1)$.

   3.2.2 $opt(i)\leftarrow opt(i-1)$

4. החזר את $A(n)$.

סיבוכיות: $O(n\log n)$.

מילה בינארית רחוקה:

מרחק בין שתי מחרוזות בינאריות באורך n: $x=x_1x_2\cdots x_n$ ו- $y=y_1y_2\cdots y_n$ מוגדר להיות:
$d(x,y)=\left|\{i\in[n]:x_i\ne y_i\}\right|$.

המרחק של מחרוזת w מקבוצת מחרוזות A נתון ע"י: $d(w,A)=\min\{d(w,a):a\in A\}$.

הרעיון: לבנות את w בצורה חמדנית- את w_i נקבע לערך ההפוך מהביט ה- i המתקבל ע"י רוב המחרוזות ב- A אשר קרובות ביותר ל- w כרגע.

האלגוריתם:

קלט: קבוצה A של k מחרוזות בינאריות מאורך n.

פלט: מחרוזת בינארית w כך ש- $d(w,A)=\Omega(n/\log k)$.

1. $B \leftarrow A$.
2. עבור על i מ-1 עד- n:

2.1 חשב כמה וקטורים יש ב- B כך שהכניסה ה- i-ית שלהם היא 0 וכמה כאלה יש עם 1 בכניסה זו. אם יש יותר וקטורים עם כניסה 0 קבע $w_i \leftarrow 1$, אחרת: קבע: $w_i \leftarrow 0$.

2.2 הסר מ- B את כל הוקטורים שהכניסה ה- i-ית שלהם שונה מהכניסה של w_i.

2.3 אם B ריקה קבע $B \leftarrow A$.

סיבוכיות: $O(nk)$.

צביעה של גרפים:

צביעה חוקית: יהא $G=(V,E)$ גרף לא מכוון. צביעה חוקית של G היא פונקציה $c:V \to N$ כך שלכל $uv \in E$ מתקיים: $c(u) \neq c(v)$.

המספר הכרומאטי: המספר המינימאלי של צבעים הדרוש לצביעה של גרף G. סימון: $\chi(G)$.

נסמן ב- $d_1 \geq d_2 \geq \cdots \geq d_n$ את הדרגות של צמתי הגרף, מסודרים בסדר לא עולה.

האלגוריתם:

קלט: גרף לא מכוון $G=(V,E)$ וסדר כלשהו על הצמתים.

פלט: צביעה חוקית c של G המשתמשת בלכל היותר d_1+1 צבעים.

1. עבור צומת עם הסדר הנתון של הצמתים. יהא v הצומת הנוכחי.
2. קבע את $c(v)$ להיות הצבע המינימאלי שאינו בשימוש ע"י שכניו של v.

סיבוכיות: $O(V+E)$.

אלגוריתם חמדן משופר:

קלט: גרף לא מכוון $G=(V,E)$.

פלט: צביעה של G בלכל היותר $d_{\chi(G)}+1$ צבעים.

1. מיין את צמתי הגרף לפי דרגותיהם, מהגבוה לנמוך.
2. הרץ את האלגוריתם הקודם עם הצמתים מסודרים לפי המיון משלב (1).

סיבוכיות: $O(V+E\log E)$.

הערה: האלגוריתם הוכיח אי-שוויון קומבינטורי מעניין: $\chi(G) \leq d_{\chi(G)}+1$.

שברים מצריים:

טענה: כל מספר רציונאלי בתחום $(0,1)$ ניתן לכתיבה כסכום סופי של שברים מצריים שונים.

האלגוריתם: כל עוד אין בידינו דבר מצרי, מצא את השבר המצרי הגדול ביותר שנכנס בתוך (כלומר, הקטן מ) השבר שבידינו- שבר זה יהיה אחד המחוברים. על ההפרש נריץ את האלגוריתם ברקורסיה.

כיסוי בצמתים של עצים:

כיסוי בצמתים (Vertex Cover): בהינתן גרף לא מכוון $G=(V,E)$, כיסוי בצמתים של G הוא קבוצת צמתים $U\subseteq V$ כך שלכל קשת $uv\in E$, לפחות אחד מהצמתים u,v שייך ל- U.

הרעיון: נביט בעלה v של העץ. יהיה u השכן (היחיד) שלו. כל כיסוי חייב לכסות את הקשת uv ולכן בהכרח מכיל אחד מבין u,v. נבחין כי אין טעם לקחת את v לכיסוי שכן v מכסה אך ורק את uv וזאת לעומת u שאולי מכסה גם קשתות נוספות.

האלגוריתם:

קלט: עץ לא מכוון $G=(V,E)$.

פלט: כיסוי צמתים ל- G

1. $U\leftarrow\varnothing$.
2. כל עוד לא קיים עלה v ב- G:

    2.1 הוסף את u, השכן של v ל- U.

    2.2 הסר את כל הקשתות הנוגעות ב- u.

סיבוכיות: $O(V+E)$.

טענה-1: בכל יער בו יש קשת אחת לפחות, יש שני עלים. ולכן כל עוד יש קשתות בגרף, יש עלה.

טענה-2: לכל i קיים כיסוי מינימאלי המכיל את הצמתים $v_1,v_2,\cdots,v_i$ (סדר הכניסה ל- U).

קידוד הופמן:

טענה: אם כל תו מופיע בקובץ בתדירות קטנה ממש מ- $1/3$ אזי בהכרח בעץ הופמן לא תהיה מילת קוד מאורך 1.

טענה: לכל אלגוריתם כיווץ ולכל אורך n טבעי, קיימת מילה $x\in\{0,1\}^n$ שהאלגוריתם אינו מכווץ.

תכנות דינאמי:

סידור אופטימאלי לכפל של n מטריצות:

נתונה סדרה של n מטריצות: $A_1, A_2, \cdots, A_n$ ונרצה לחשב את המכפלה: $A_1 \times A_2 \times \cdots \times A_n$. הסדר בו נבצע את ההכפלות ישפיע במידה רבה על סיבוכיות החישוב.

האלגוריתם:

קלט: n מטריצות $A_1, \cdots, A_n$ כאשר מטריצה A_i ממימד $p_{i-1} \times p_i$.

פלט: סידור סוגריים אופטימאלי למכפלה $A_1 \times \cdots \times A_n$.

נחזיק מערך דו-מימדי $m[i, j]$, שישמור בכניסה ה- i, j את מספר ההכפלות המינימאלי בהכפלת $A_i \times \cdots \times A_j$. ונרצה לקבל כפלט את $m[1, n]$.

נחשב את $m[i, j]$ באופן רקורסיבי:

- אם $i = j$, אז השרשרת מכילה מטריצה בודדת ולכן $m[i, j] = 0$.

- אם $i < j$, נניח שיש השמת סוגריים אופטימאלית שמפצלת את $A_i, \cdots, A_j$, בין A_k ל- A_{k+1}, אז מספר המכפלות עבור $A_i, \cdots, A_j$ הוא: $m[i, j] = \underbrace{m[i,k]}_{B_1} + \underbrace{m[k+1, j]}_{B_2} + \underbrace{p_{i-1} \times p_k \times p_j}_{B_1 \times B_2}$. כעת יש $(j-1)$ אפשרויות לבחירת k ולכן:

$$m[i, j] = \begin{cases} 0 & i = j \\ \min\limits_{i \le k \le j} \left\{ m[i,k] + m[k+1] + p_{i-1} \times p_k \times p_j \right\} & i < j \end{cases}$$

1. עבור $1 \le i \le n$ בצע: $m[i,i] \leftarrow 0$.

2. עבור $j = 1$ עד $n-1$ בצע: (אורך הרצפים נקבע ע"פ j).

2.1 עבור $l = 1$ עד $n-j$ בצע: (המטריצה השמאלית ביותר נקבעת לפי l)

2.1.1 $m[l, l+j] \leftarrow \min\limits_{l \le k \le l+j} \left\{ m[l,k] + m[k+1, l+j] + p_{l-1} p_k p_{l+j} \right\}$.

סיבוכיות: $O\left(n^3\right)$.

אלגוריתם פלויד וורשל למציאת כל המסלולים הקלים ביותר:

פונקצית המשקל המוכללת: $\bar{w}(i,j)=\begin{cases}0 & i=j\\ w(i,j) & (i,j)\in E\\ \infty & otherwise\end{cases}$

פונקצית חסם עליון המוכללת: פו' $d:V\times V\to R$ שמקיימת: $dist(i,j)\le d(i,j)\le \bar{w}(i,j)$.

כלל שיפור מקומי: עבור השלשה (i,k,j): אם $d(i,k)+d(k,j)<d(i,j)$ אז:
$d(i,j)\leftarrow d(i,k)+d(k,j)$.

טענה: אם d פונקצית חסם עליון מוכללת והופעל עליה כלל שיפור מקומי, אז אחרי העדכון d נותרה פונקצית חסם עליון מוכללת.

האלגוריתם:

קלט: $G=(V,E)$, פונקצית משקל $w:E\to R$ ובה"כ: $V=\{1,2,\cdots,n\}$.

פלט: לכל זוג צמתים לכל זוג צמתים $(i,j)\in V$ - $d(i,j)=dist(i,j)$.

1. לכל זוג צמתים (i,j) בצע: $d(i,j)\leftarrow \bar{w}(i,j)$.
2. עבור $k=1$ עד n בצע:

   2.1 לכל זוג צמתים i,j בצע שיפור (i,k,j) כלומר:
   $d(i,j)=\min_k\{d(i,j),d(i,k)+d(k,j)\}$.

סיבוכיות: $O(V^3)$.

רמה-k: מסלולים שמכילים בתוך צמתים פנימיים רק צמתים מ-1 עד- k.

$dist^{(k)}(i,j)$: משקל המסלול המינימאלי מבין כל המסלולים ברמה ה- k מ- i ל- j או ∞ אם אין מסלול כזה.

למה-1: לכל $o\le k\le n$, אחרי איטרציה k מתקיים: $d(i,j)\le dist^{(k)}(i,j)$.

טענה: לכל $1\le k\le n$: $dist^{(k)}(i,j)=\min\begin{cases}dist^{(k-1)}(i,j)\\ dist^{(k-1)}(i,k)+dist^{(k-1)}(k,j)\end{cases}$.

:Knapsack

נתון תרמיל גב שלו קיבולת משקל W ונתונים n עצמים $a_1,\cdots,a_n$ עם משקלים חיוביים $w_1,\cdots,w_n$. לכל עצם a_i מותאם מספר p_i המציין את החשיבות של העצם. רוצים להכניס לשק קבוצת עצמים שחשיבותם מקסימאלית. כלומר:

$$\sum_{1\le i\le n} w_i \le W$$

$$\sum_{1\le i\le n} p_i = \max\left\{\sum p_j \quad and \quad \sum w_j \le W\right\}$$

נסמן ב- $F(k,w)$ את הרווח המקסימאלי כאשר בוחרים עצמים מתוך $a_1,\cdots,a_k$ בלבד, וגודל התרמיל הוא w. אנו רוצים לחשב את $F(n,W)$.

טענה-1: אם $k=0$ או $w\le 0$ אז $F(k,w)=0$. אחרת:

$$F(k,w)=\begin{cases} F(k-1,w) & w_k > w \\ \max\left\{F(k-1,w),\, p_k + F(k-1,w-w_k)\right\} & otherwise \end{cases}$$

כלומר: אם המשקל חוקי, נרצה את המקסימאלי מבין רצף ההכנסות שכולל את הפריט ורצף ההכנסות שלא כולל את הפריט.

האלגוריתם:

צור מטריצה A מגודל $(n+1)\times(W+1)$ ומלא את השורה המתאימה ל $k=0$ עד לשורה המתאימה ל- $k=n$, כאשר כל שורה מלא בסדר כלשהו. את חישוב כניסות המטריצה בצע לפי הנוסחה שבטענה-1. בסיום החזר את $A[n,W]$.

סיבוכיות: $O(nW)$ (כל כניסה ב- $O(1)$). סיבוכיות פסאודו פולינומיאלית.

הערה: ניתן למצוא מהו הפיתרון (ולא רק את ערכו) ע"י סימון איזו כניסה עדכנה את הכניסה הנוכחית במטריצה.

קבוצה בלתי תלויה מקסימאלית בעץ:

מציאת קבוצה בלתי תלויה מקסימאלית בעץ מכוון $T=(V,E)$.

הרעיון: נכוון את העץ ונסמן ב- r את השורש. ולכל צומת v נסמן:

- $S^+(v)$: גודל קבוצה ב"ת מקסימאלית בתת עץ ש- v שורשו הכוללת את v.
- $S^-(v)$: גודל קבוצה ב"ת מקסימאלית בתת עץ ש- v שורשו שאינה כוללת את v.
- $S(v)$: גודל קבוצה ב"ת מקסימאלית בתת עץ ש- v שורשו, ללא הגבלות.

טענה-2: לכל $v\in V$ מתקיים:

$$S(v)=\max\left\{S^+(v),S^-(v)\right\}$$

$$S^+(v)=1+\sum_{u\in child(v)} S^-(u)$$

$$S^-(v)=\sum_{u\in child(v)} S(u)$$

האלגוריתם:

קלט: עץ לא מכוון $T=(V,E)$.

פלט: גודל קבוצה בלתי תלויה מקסימאלית.

1. הרץ DFS על T עם הווריאציה הבאה: בכל פעם שמבצעים נסיגה מצומת v, חשב את $S^-(v), S^+(v), S(v)$.
2. החזר את $S(r)$.

סיבוכיות: $O(V+E)$.

קבוצה בלתי תלויה מקסימאלית בגרף כללי:

האלגוריתם:

1. $S \leftarrow \varnothing$.
2. כל עוד $V \neq \varnothing$:

   2.1 בחר צומת $v \in V$ בעל דרגה מינימאלית d בגרף הנוכחי.

   2.2 הוסף את v ל- S.

   2.3 מחק את v ואת כל שכניו_עם הקשתות החלות עליהם) מ- G.
3. החזר את S.

סיבוכיות: $O(V+E)$

משפט: האלגוריתם מחזיר קבוצה ב"ת עם $\frac{n}{\delta+1}$ צמתים, כאשר δ זו הדרגה המינימאלית בגרף.

תת-סדרה מונוטונית ארוכה ביותר:

תהא $a_1, a_2, \cdots, a_n$ סדרת מספרים. נחפש תת-סדרה מונוטונית לא יורדת ארוכה ביותר של הסדרה.

הרעיון: נמצא לכל k, אורך תת-סדרה מונוטונית לא יורדת ארוכה ביותר של הסדרה המסתיימת ב- a_k. כלומר הערך המבוקש הוא: $\max\{T(k) : k \in n\}$.

$A(k)=\{j : j<k \ and \ a_j \le a_k\}$ - קבוצת האינדקסים של כל האיברים שיכולים להופיע לפני a_k בתת-סדרה מונוטונית.

טענה:
$$T(k)=1+\begin{cases} \max\{T(j) : j \in A(k)\} & A(k) \neq \varnothing \\ 0 & A(k)=\varnothing \end{cases}$$

סיבוכיות: כל ערך $T(k)$ ניתן לחשב ב- $O(k)$ ולקיחת מקסימום על כולם תיתן: $O(n^2)$.

בעיית הסוכן הנוסע האוקלידית הביטונית:

נתונות n נקודות במישור, יש לחשב את המעגל הקצר ביותר המחבר את כל הנקודות.

מעגל ביטוני: מסלול המתחיל בנקודה השמאלית ביותר, מתקדם ימינה דרך חלק מהנקודות ואז חוזר שמאלה דרך כל הנקודות הנותרות.

הרעיון: נסמן את הנקודות ב- $p_1, p_2, \cdots, p_n$ לפי קואורדינאטת x עולה. ניסוח שקול לבעיה: שני סוכנים שרוצים לנוע מ- p_1 ל- p_n תוך שהם נעים ימינה בלבד, יחדיו מבקרים בכל הנקודות אך אינם מבקרים באותה נקודה. מציאת זוג מסלולים כאלה משרה מסלול ביטוני.

$d(i, j)$ - המרחק בין הנקודות p_i, p_j.

$B(i)$ - אורך מסלול ביטוני קצר ביותר עבור הנקודות $p_1, \cdots, p_i$. נרצה לחשב את $B(n)$.

$B(i, j)$ - נתנה את המסלול בנקודה- j שבה מבקר הסוכן לפני שביקר בנקודה האחרונה- i.

$$B(i, j) = \underbrace{B(j+1)}_{length\ whrn\ both\ finish\ at\ j+1} - \underbrace{d(j, j+1)}_{no\ one\ visit\ both\ j\ \&\ j+1} + \underbrace{d(j, i)}_{distance\ for\ 1st\ 'till\ i} + \underbrace{\sum_{k=j+1}^{i-1} d(k, k+1)}_{distance\ for\ 2nd:\ all\ dots\ from\ j+1\ to\ i}$$

ואז החישוב המבוקש יהיה: $B(i) = \min\{B(i, j) : j \in [i-2]\}$

סיבוכיות: $O(n^3)$.

רווח מקסימאלי בתורת המשחקים:

משחק שבו הרווח המקסימאלי של כל שחקן הינו סכום מספריו פחות סכום השחקן השני.

הרעיון:

עבור $i \le j$ נסמן ב- $T(i, j)$ את הרווח של אליס עבור משחק על הסדרה $a_i, a_{i+1}, \cdots, a_j$. היחס הרקורסיבי הוא היחס הבא:

$$T(i, j) = \begin{cases} a_i & i = j \\ \max\{a_i - T(i+1, j), a_j - T(i, j-1)\} & i \ne j \end{cases}$$

אם $i = j$: יש רק איבר אחד בסדרה ולכן אין ברירה אלא לקחת אותו ולהרוויח a_i.

אחרת, יש בחירה בין לקחת a_i ל-לקחת a_j. ואז לפי הבחירה, התור עובר לשחקן השני והסכום שהוא הרוויח עקב בחירה זו, יורק מסכום השחקן הראשון ולכן המינוס.

סיבוכיות: $O(n^2)$.

רשתות זרימה וזרימת מקסימום:

אפיוני רשתות זרימה:

- קיבולת על הקשתות שחוסמת את כמות הזרימה על כל קשת.
- צומת מקור שמייצר זרימה.
- צומת בור (או "יעד") שקולט את הזרימה.
- זרימה על הקשתות.

תכונות של זרימה ברשתות:

נסמן ב- $f(e)$ את הזרימה על הקשת המכוונת $e \in E$.

אילוץ הקיבול: הזרימה דרך הקשת e אינה יכולה לחרוג מקיבול הקשת, כלומר: $0 \le f(e) \le c(e)$.

שימור הזרימה: לכל הצמתים $v \in V \setminus \{s,t\}$, סה"כ הזרימה שנכנסת ל- v שווה לסה"כ הזרימה שיוצאת מ- v.

ערך פונקצית הזרימה: סה"כ הזרימה מהמקור לבור, סימון: $|f|$ או F

$F = \sum_{e \in in(t)} f(e) - \sum_{e \in out(t)} f(e)$.

חתך s-t ברשת זרימה:

חתך $(S,\bar{S})$ כך ש- $S \subset V,\ \bar{S} = V \setminus S$ ומתקיים: $s \in S, t \in \bar{S}$.

קיבול של חתך $(S,\bar{S})$: $c(S,\bar{S}) = \sum_{e \in (S,\bar{S})} c(e)$ (סכום קשתות שיוצאת מ S ל- $\bar{S}$).

למה-1: לכל חתך s-t $(S,\bar{S})$ ולכל פונקצית זרימה f מתקיים: $F = \sum_{e \in (S,\bar{S})} f(e) - \sum_{e \in (\bar{S},S)} f(e)$.

מסקנה: אפשר לחשב את ערך פונקצית הזרימה F ע"י הסתכלות על חתך כלשהו $(S,\bar{S})$, או ע"י חישוב סה"כ הזרימה מהמקור או ע"י חישוב סה"כ הזרימה לבור.

למה-2: לכל חתך $(S,\bar{S})$ ולכל פונקצית זרימה f מתקיים: $F \le c(S,\bar{S})$.

מסקנה: אם קיים חתך s-t $(S,\bar{S})$ כך ש- $F = c(S,\bar{S})$ אז F היא זרימת מקסימום.

מסלולי שיפור:

קיבול שיורי: לכל קשת $e = (u,v)$ עם קיבול $c(e)$ וזרימה $f(e)$ נגדיר שתי קשתות עם קיבול שיורי:

1. קשת קדמית (u,v) עם קיבול $c(e) - f(e)$.
2. קשת אחורית (v,u) עם קיבול $f(e)$.

הערה: למעשה, קיבול כל קשת הינו סך הזרימה שניתן להזרים עליה בנוסף לזרימה הקיימת.

הגרף השיורי: יסומן G_f הוא אוסף כל הקשתות עם קיבול שיורי חיובי ממש.

מסלול שיפור: מסלול מ- s ל- t בגרף השיורי.

למה-3: עבור מסלול שיפור p תהי Δ הקיבול השיורי הקטן ביותר לאורך p. אז א נגדיל את הזרימה ב- Δ בכל קשת המתאימה לקשת קדמית ב- p ונטין את הזרימה ב- Δ בכל קשת המתאימה לקשת אחורית ב- p, נקבל פונקציית זרימה חוקית כך שערך הזרימה גדל ב- Δ.

משפט חתך מינימום-זרימת מקסימום (Min Cut – Max Flow):

תהי f פונקציית זרימה ברשת זרימה $N=(G,s,t,c)$. אזי הטענות הבאות שקולות:

1. f זרימת מקסימום.
2. אין מסלול שיפור מ- s ל- t ברשת השיורית.
3. קיים חתך s-t $(S,\bar{S})$ עבורו $F=c(S,\bar{S})$ (וזהו חתך מינימום).

טענה-1: אם $e\in(S,\bar{S})$ אז $f(e)=c(e)$. כלומר, e רוויה.

טענה-2: אם $e\in(\bar{S},S)$ אזי $f(e)=0$.

האלגוריתם של Ford-Fulkerson למציאת זרימת מקסימום:

האלגוריתם הגנרי:

1. אתחול- זרימה 0 בכל הקשתות.
2. כל עוד קיים מסלול שיפור p מ- s ל- t ב- G_f, שפר עליו את הזרימה (ניתן למצוא מסלול שיפור ע"י הרצת BFS, למשל).

הערות:

- אם הקיבולים על הקשתות לא רציונאליים, לא מובטח שהאלגוריתם יעצור.
- אם הקיבולים על הקשתות שלמים, האלגוריתם ימצא זרימת מקסימום בשלמים, שהרי יש שיפור ב-1 לפחות בכל איטרציה. נשים-לב שמספר האיטרציות עלול להגיע עד ל- F^*, ערך זרימת המקסימום.

סיבוכיות: מספר האיטרציות: $O(|f^*|)$, זמן לביצוע כל איטרציה: $O(V+E)$. סה"כ: $O(f^*E)$.

האלגוריתם של אדמונדס-קרפ:

בכל איטרציה בוחרים את מסלול השיפור הקצר ביותר מ- s ל- t.

גרעין של רשת זרימה: אוסף קשתות ב- (G,s,t,c) על מסלול קצר ביותר מ- s ל- t. סימון: $S(G)$.

טענה-1: תהי (G,s,t,c) רשת זרימה ותהי $(u\to v)\in S(G)$ אז הוספת הקשת $(v\to u)$ ל- G אינה משנה את הגרעין $S(G)$.

מסקנה: אם נוסיף k קשתות שהפוכות לקשתות גרעין, הגרעין יישמר.

טענה-2: יהיו G' ו- G'' שתי רשתות שיוריות באיטרציות עוקבות של האלגוריתם. אז לפחות אחד מהשניים מתקיים:

1. $\delta_{G'}(s,t)<\delta_{G''}(s,t)$ (המסלול הקצר ביותר מתארך).
2. $|S(G'')|<|S(G')|$ (גודל הגרעין קטן).

מסקנה: מקרה-1 יכול להתרחש $|V|$ פעמים לכל היותר. בין שתי פעמים עוקבות שבהן 1 התרחש, הגרעין יכול היה לקטון בלכל היותר $|E|$. ולכן כמות האיטרציות היא: $O(VE)$.

סיבוכיות: איטרציה בודדת לוקחת: $O(E)$ ולכן בסה"כ: $O(E^2V)$.

האלגוריתם של דיניץ:

גרף שכבות: בהינתן גרף שיורי G_f, גרף השכבות- L_f הוא תת-גרף של הגרף השיורי, המכיל רק את הקשתות המעבירות אותנו משכבה אחת לאחרת ועוצר כשמגיע ל- t.

זהו תת-גרף המכיל את כל המסלולים הקצרים ביותר מ- s ל- t. סימון: $\delta_f(v)$ - אורך המסלול הקצר ביותר מ- s ל- v.

מכיל רק קשתות $v \to u$ ב- G_f עבורן: $\delta_f(v) = \delta_f(u) + 1$.

הגרף מכיל רק צמתים v כך ש- $\delta_f(v) \le \delta_f(t)$.

זרימה חוסמת בגרף השכבות: זרימה שאין עבורה מסלולי שיפור בגרף השכבות (כלומר, אין מסלולים מכוונים מ- s ל- t בגרף השכבות לאחר שמוחקים קשתות עם קיבול אפס).

האלגוריתם:

1. $f(e) \leftarrow 0$ לכל קשת $e \in E$.
2. כל עוד רשת השכבות (L_f, s, t, c_f) מכילה מסלול מ- s ל- t:
   2.1 מצא זרימה חוסמת f' ברשת השכבות.
   2.2 $f \leftarrow f + f'$ (שיפור f באמצעות הזרימה החוסמת).
3. החזר את f.

טענה-1: בסיום ריצת האלגוריתם, f היא זרימת מקסימום של (G, s, t, c).

טענה-2: יהיו G' (הראשונה) ו- G'' (השנייה) שתי רשתות שיוריות באיטרציות עוקבות של האלגוריתם של דיניץ. אז מתקיים: $\delta_{G'}(s,t) < \delta_{G''}(s,t)$.

מסקנה: האלגוריתם של דיניץ מבצע לכל היותר $|V|$ איטרציות.

מציאת זרימה חוסמת:

1. $f' \leftarrow 0$ לכל e ברשת השכבות.
2. כל עוד יש קשתות שיוצאות מ- s:
   2.1 הרץ DFS מ- s עד שתגיע לצומת עם דרגת יציאה 0.
   2.2 אם הצומת הזה אינו t, מוחקים את כל הקשתות שנכנסות אליו.
   2.3 אם הצומת הזה הוא t:
      2.3.1 יהי p המסלול מ- s ל- t בסריקה הנוכחית.
      2.3.2 מוסיפים ל- f' את הזרימה הגדולה ביותר שניתן להזרים על-פני p.
      2.3.3 מוחקים את כל הקשתות ב- p שלא ניתן להזרים עליהן עוד זרימה.

סיבוכיות: $O(VE)$.

זרימה עם חסמים תחתונים:

רשת זרימה עם קיבולים עליונים ותחתונים היא: (G,s,t,c,b). $b:E\rightarrow R^{+}$ מציין את הקיבולים התחתונים על כל קשת.

<u>הערה:</u> לא תמיד קיימת זרימה חוקית, בניגוד לזרימה ללא חסמים תחתונים בה תמיד יש זרימה אפס שהיא חוקית.

הכרעה האם יש זרימה חוקית:

נבנה רשת זרימה $N'=(G',s',t',c')$ בעלת קיבולים עליונים בלבד.

$V'=V\cup\{s',t'\}$, $E'=E\cup\{t\rightarrow s, s\rightarrow t\}\cup\{s'\rightarrow u, u\rightarrow t' : u\in V\}$,

ועבור כל $e\in E$: $c'(e)=c(e)-b(e)$ ו- $c'(t\rightarrow s)=\infty$.

$$c'(s'\rightarrow u)=\sum_{e\in in(e)} b(e)$$

$$c'(u\rightarrow t')=\sum_{e\in out(u)} b(e)$$

<u>משפט:</u> קיימת זרימה חוקית ב- N אם"ם קיימת ב- N' זרימת מקסימום שערכה הוא: $\sum_{e\in E} b(e)$.

שידוך מקסימום בגרף דו-צדדי:

עבור $G=(V=L\cup R,E)$ גרף דו-צדדי:

שידוך: תת-קבוצה של קשתות M כך שלכל שתי קשתות אין נקודת קצה משותפת.

שידוך מקסימום: שידוך M כך שלכל שידוך M' אחר מתקיים: $|M'|\le|M|$.

שידוך מושלם: שידוך M כך שכל צומת בגרף היא נקודת קצה של איזושהי קשת מ- M.

מציאת שידוך מקסימום בעזרת זרימה:

נגדיר את הגרף המכוון $G'=(V',E')$ באופן הבא:

$$V'=V\cup\{s,t\}$$

$$E'=\{(s,u):u\in L\}\cup\{(u,v):uv\in E,u\in L,v\in R\}\cup\{(v,t):v\in R\}$$

טענה-1: ב- G יש שידוך בגודל k אם"ם ב- G' יש זרימה בגודל k.

האלגוריתם:

קלט: גרף דו-צדדי $G=(V=L\cup R,E)$.

פלט: שידוך מקסימום ב- G.

1. בנה את רשת הזרימה $N'=(G',s,t,c)$ עם G' כפי שהוגדר לעיל ו- $c\equiv 1$.
2. הרץ את פורד-פולקרסון לחישוב זרימת מקסימום ב- N'.
3. החזר: $M=\{uv: f(u,v)=1\}$.

סיבוכיות: $O(VE)$.

כיסוי מינימאלי במסלולים:

כיסוי במסלולים: יהי $G=(V,E)$ גרף מכוון. כיסוי במסלולים של צמתי G הוא קבוצת מסלולים זרים בצמתים ב- G כך שכל צומת ב- G מוכל באחד המסלולים בכיסוי (מסלול עשוי להיות מורכב מצומת יחיד).

הרעיון: לייצג מסלול ע"י כך שנשדך לכל צומת את הצומת העוקב לו במסלול.

נגדיר גרף דו-צדדי לא מכוון $G'=(V'=L\cup R,E')$ באופן הבא:

$$L=\{v_1:v\in V\}\quad R=\{v_2:v\in V\}\quad E'=\{u_1v_2:(u,v)\in E\}$$

טענה-2: ב- G יש כיסוי במסלולים בגודל k אם"ם ב- G' יש שידוך בגודל $n-k$.

האלגוריתם:

קלט: גרף $G=(V,E)$ DAG.

פלט: כיסוי מינימאלי במסלולים.

1. חשב את G'.
2. מצא שידוך מינימום M ב- G'.
3. החזר כיסוי שקשתותיו הן כל הקשתות uv כך ש- $u_1v_2\in M$.

סיבוכיות: $O(VE)$.

משפט החתונה של Hall:

הגדרה: יהא $G=(V,E)$ גרף לא מכוון. בהינתן קבוצת צמתים A, נסמן ב- $\Gamma(A)$ את קבוצת השכנים של צמתי A, כלומר: $\Gamma(A)=\{v\in V : \exists u\in A, uv\in E\}$.

המשפט: יהא $G=(V=L\cup R,E)$ גרף דו-צדדי עם $|L|=|R|=n$. ב- G קיים שידוך מושלם אם"ם לכל $A\subseteq L$ מתקיים: $|\Gamma(A)|\geq|A|$.

הקשירות בקשתות:

קשירות בקשתות: בהינתן גרף לא מכוון $G=(V,E)$, הקשירות בקשתות של G, הוא המספר המינימאלי של קשתות שהסרתן מ- G הופכת אותו ללא קשיר. סימון: $\kappa(G)$.

טענה-1: נסמן ב- $e(A)$ את מספר הקשתות החוצות חתך $(A,V\setminus A)$ כלשהו בגרף (זהו אינו חתך s-t). אז מתקיים: $\kappa(G)=\min\{e(A) : A\subseteq V, A\notin\{\varnothing, V\}\}$, כלומר: מספר הקשתות המינימאלי בחתך כלשהו בגרף.

הגדרת רשת זרימה עבור הבעיה:

נגדיר גרף מכוון $G'=(V,E')$ באופן הבא: $E'=\{(u,v),(v,u) : uv\in E\}$. ניתן לכל קשת קיבול יחידה ונקבע את s להיות צומת שרירותי מ- V.

טענה-2: לכל $t\in V$ נסמן ב- f_t^* הזרימה המקסימאלית ברשת $N_t=(G',s,t,c)$. אז מתקיים: $\kappa(G)=\min\{f_t^* : t\in V\}$.

האלגוריתם:

קלט: גרף לא מכוון וקשיר $G=(V,E)$.

פלט: $\kappa(G)$.

1. חשב את G'.
2. קבע s באופן שרירותי, ולכל $t\neq s$ חשב את f_t^*.
3. החזר: $\min\{|f_t^*| : t\in V\setminus\{s\}\}$.

סיבוכיות: להרצה אחת לחישוב זרימת מקסימום: $O(Ef^*)=O(VE)$ (כי $|f_t^*|<n$ לכל t). ול- n הרצות: $O(V^2E)$.

קונבולוציות והתמרת פורייה המהירה (FFT):

כפל פולינומים:

נתון פולינום ממעלה n $(a_n \neq 0)$, $A(x) = \sum_{j=0}^{n} a_j x^j$.

ייצוג-1: שמירת הפולינום בעזרת וקטור המקדמים שלו- $\bar{a} = (a_0, a_1, \cdots, a_n)$.

חישוב ערך בנקודה: $O(n)$; חיבור שני פולינומים: $O(n)$; כפל שני פולינומים: $O(n^2)$

ייצוג-2: פולינום ממעלה n ניתן לייצוג בעזרת $n+1$ נקודות שמקיימות: $y_i = A(x_i)$.

חיבור שני פולינומים (עבור אותן נקודות x): $O(n)$; סיבוכיות הכפל: $O(n)$.

מעבר מייצוג ע"י וקטור לייצוג ע"י נקודות:

בוחרים $n+1$ ערכי x שונים, מציבים ומקבלים את ערכי y המתאימים.

מעבר מייצוג ע"י נקודות לייצוג ע"י וקטור- משפט האינטרפולציה:

בהינתן $n+1$ נקודות שונות (x_i, y_i) כך ש- $x_i \neq x_j$ אז קיים פולינום יחיד A ממעלה- n, כך ש- $A(x_i) = y_i$.

סיבוכיות מעבר בין הייצוגים: $O(n \log n)$.

אלגוריתם יעיל לכפל פולינומים – קונבולוציה:

1. נבחר $2n$ ערכים $x_1, \cdots, x_{2n}$ ונמצא את ערכי הפולינומים $A(x_j), B(x_j)$ עבור $1 \leq j \leq 2n$.
2. נחשב את $C(x_j) = A(x_j) \cdot B(x_j)$ לכל j.
3. נחשב את המקדמים של $C(x)$- $c_0, c_1, \cdots, c_{2n-2}$ מהערכים שחישבנו- $C(x_1), \cdots, C(x_{2n})$ תוך שימוש במשפט האינטרפולציה.

הבעיה: בשלב (1): חישוב ישיר של ערך הפונקציה ממעלה $n-1$ ב- $2n$ נקודות דורש $\Omega(n^2)$ ולכן נתגבר על הבעיה ע"י לקיחת $2n$ נקודות שיש ביניהן קשר: שורשי היחידה המרוכבים.

למה-1: ניתן לחשב את $A(x)$ ב- $2n$ שורשי היחידה ב- $O(n \log n)$ צעדים.

למה-2: לכל פולינום $C(x) = \sum_{s=0}^{2n-1} c_s x^s$ והפולינום המתאים $D(x) = \sum_{s=0}^{2n-1} d_s x^s$ כאשר $d_s = C(w_{s,2n})$, מתקיים: $c_s = \frac{1}{2n} D(w_{2n-s,2n})$ (כלומר נוכל לקבל את מקדמי הפולינום $C(x)$ ע"י חישוב ערכי $D(x)$ בשורשי היחידה).

חיבור איברי קבוצות ביעילות:

יהיו $A,B\subseteq\{1,2,\cdots,n\}$. נגדיר: $A+B=\{a+b:a\in A,b\in B\}$.

אלגוריתם לחישוב A+B:

קלט: קבוצות A,B ומספר n כך ש- $A,B\subseteq\{1,2,\cdots,n\}$.

פלט: הקבוצה $A+B$.

1. חשב את הפולינומים: $b(x)=\sum_{i=1}^{n}b_ix^i$, $a(x)=\sum_{i=1}^{n}a_ix^i$ כאשר

$$b_i=\begin{cases}1 & i\in B\\ 0 & i\notin B\end{cases}\qquad a_i=\begin{cases}1 & i\in A\\ 0 & i\notin A\end{cases}$$

2. חשב את הפולינום: $(ab)(x)$ כאשר עבור כל $1\le k\le 2n$, המקדם של x^k בפולינום ab הוא: $c_k=\sum_{i=1}^{2n}a_ib_{k-i}$. אך כך $c_k\ge 1$ אם"ם $k\in A+B$.

3. החזר: $\{i:c_k\ge 1\}$.

סיבוכיות: $O(n\log n)$.

מחזור מינימאלי של מחרוזת בינארית:

תהא $s=s_0s_1\cdots s_{n-1}$ מחרוזת בינארית.

מחזור של מחרוזת: נאמר כי s היא בעלת מחזור t אם לכל $i\equiv_t j$ (מודולו t) מתקיים $s_i=s_j$.

סימון: $T(s)$ - המספר המינימאלי t שמקיים תכונות מחזור של מחרוזת.

האלגוריתם:

קלט: מחרוזת בינארית $s=s_0s_1\cdots s_{n-1}$.

פלט: $T(s)$.

1. חשב את הפולינומים

$$p(x)=(-1)^{s_0}+(-1)^{s_1}x+(-1)^{s_2}x^2+\cdots+(-1)^{s_{n-1}}x^{n-1}$$
$$q(x)=(-1)^{s_{n-1}}+(-1)^{s_{n-2}}x+\cdots+(-1)^{s_0}x^{n-1}$$

2. חשב את הפולינום: $pq(x)$, כאשר עבור $1\le k\le n-1$ המקדם של x^{n-k-1} בפולינום הוא:

$$c_k=\sum(-1)^{s_1}(-1)^{s_{n-1-(n-k-1-i)}}=\cdots=(n-k)-2\cdot|\{i:s_i\ne s_{k+i}\ or\ 0\le i\le n-k-1\}|$$ ולכן

$c_k=n-k$ אם"ם ל- s יש מחזור k (ואחרת $c_k<n-k$).

3. החזר את ה- k המינימאלי עבורו $c_k=n-k$ ואם לא קיים כזה, החזר: n.

סיבוכיות: $O(n\log n)$.

תרגילים לדוגמא:

שאלה 4 (30 נקודות)

נתונים n אינטרוולים $I_1, I_2, \ldots, I_n$ בקטע $[0,1]$ כך שלכל $x \in [0,1]$ קיים j כך ש: $x \in I_j$.

הגדרה: כיסוי היא תת-קבוצה של האינטרוולים $I \subseteq \{I_1, I_2, \ldots, I_n\}$ כך שלכל $x \in [0,1]$ קיים אינטרוול ב- I המכיל את x.

בכל אחד משני הסעיפים הבאים יש להציע אלגוריתם יעיל ככל שניתן הפותר את הבעיה, להוכיח את נכונותו ולנתח את סיבוכיותו.

א. יש למצוא כיסוי קטן ביותר. (15 נקודות)

ב. נתון כי לכל אינטרוול I_j מתאים משקל אי-שלילי w_j. יש למצוא כיסוי בעל סך משקלים קטן ביותר. (15 נקודות)

א. אלגוריתם:

1. נמיין את n האינטרוולים, פעם לפי נקודת התחלה ופעם לפי נקודת הסיום שלהם.
2. נאתחל $x \leftarrow 0$, $S \leftarrow \varnothing$ (S יכיל את האינטרוולים בכיסוי).
3. מצא מבין כל האינטרוולים שמתחילים לפני או בx0 את זה שמסתיים הכי רחוק, נסמנו- I_i.
4. $S \leftarrow S \cup \{I_i\}$, ונזרוק את כל האינטרוולים שעברנו עליהם והם לא I_i.
5. $x \leftarrow$ (נקודת הסיום של I_i).
6. חוזרים לשלב 1 ומפסיקים כאשר $x \geq 1$, כלומר, כאשר האינטרוול שלקחנו לכיסוי מכסה את הנקודה האחרונה בקטע.

סיבוכיות:

מיון מערך האינטרוולים פעמיים- $O(n \log n)$. עוברים על מערך האינטרוולים פעם אחת וכל איבר שעברנו זורקים ולכן המעבר ב- $O(n)$ כי בכל איבר שאנחנו עוברים, מבצעים מספר קבוע של פעולות.

נכונות:

יהי I_j האינטרוול עם נקודת הסיום הכי גדולה מבין אלה שמכילים את 0. כל כיסוי חוקי חייב להכיל את הנקודה 0. נראה כי קיים כיסוי מינימאלי שמכיל את I_i.

נניח בשלילה כי הכיסוי האופטימאלי מכיל אינטרוול אחר I_j, אזי גם $S - \{I_j\} + \{I_i\}$ הוא כיסוי חוקי כי I_i מכסה יותר נקודות מ- I_j והוא כיסוי באותו גודל. מצאנו לכן כיסוי מינימאלי שמכיל את I_i. כמו-כן, בהינתן ש- I_i שייך לכיסוי המינימאלי צריך לכסות את שאר הקטע גם ע"י כיסוי מינימאלי לכן באותה השיטה נעבור על שאר הקטעים ונמצא כיסוי אופטימאלי המכיל את כולם. ולכן האלגוריתן מחזיר כיסוי אופטימאלי.

שאלות על אינטרוולים:

סעיף ראשון נפתר ע"י אלגוריתם חמדן, מוכיחים ע"י השוואה ליפתרון אופטימאלי והחלפת האיברים בו.

סעיף שני עם משקלים ע"י תכנון דינאמי- מוכיחים באינדוקציה או ע"י גרף מסלולים קצרים.

שאלות שיש להתאים זוגות- ע"י שידוך.

א. (6 נק׳) נתון גרף מכוון ופשוט $G=(V,E)$ ושני צמתים s ו- t. ברצוננו לחשב את המספר המקסימאלי של מסלולים מכוונים זרים בקשתות בין s ל- t. נגדיר את רשת הזרימה הבאה: $N=(G,s,t,c)$ כך ש- $c(e)=1$ $\forall e\in E,$. הוכיחו כי ערך זרימת המקסימום ב N שווה למספר המקסימאלי של מסלולים מכוונים זרים בקשתות בין s ל- t.

נסמן ב-M את המספר המקסימאלי של מסלולים מכוונים זרים בקשתות בין s ל- t.

a. $|f|\leq M$

בהינתן זרימה מקסימלית עבור כל קשת בה עוברת זרימה נוסיפה למסלולים המכוונים. כלומר, אם בין u ו-v יש זרימה, אזי נאחד את u ו-v להיות באותו מסלול., ואת הקשת שלהן נאחד למסלול כמו כן. לא יתכן כי תשתתף קשת אחת בשני מסלולים, כי קיבולת הקשתות היוצאת מ-u היא 1 , כלומר סה״כ הזרימה העוברת דרכו לא יכולה להיות יתר מכמות הקשתות הנכנסות . לכן קיבלנו מסלולים זרים בקשתות, ומן הסתם זה קטן מהמספר המקסימלי האפשרי של מסלולים מכוונים זרים בקשתות בין s ל- t.

b. $|f|\geq M$

עבור כל 2 קשתות באותו מסלול ובעלות צומת משותף, כלומר: v-u-w נעביר זרימה בגודל 1 מהצומת v ל-u ומהצומת u ל-w. קיבלנו זרימה חוקית, כי קיבולת הקשתות היוצאת מ-u היא 1 , כלומר סה״כ הזרימה העוברת דרכו לא יכולה להיות יתר מכמות הקשתות הנכנסות אחרת היתה קשת המשתתפת בשני מסלולים. ולכן הזרימה החוקית לבטח קטנה מהזרימה המקסימלית האפשרית.