אלגוריתמים 1 – דפי עזר

אלגוריתמים

מיון טופולוגי

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 $V \neq \varnothing$
    1. בחר $v \in S$.
    2. קבע $L(v) \leftarrow l$
    3. קבע $l \leftarrow l+1$.
    4. הסר את v מהגרף, יחד עם כל הקשתות היוצאות ממנו.
    5. קבע $S \leftarrow S \setminus \{v\}$.
    6. הוסף ל- S את כל הצמתים מבין $\{u : vu \in E\}$ שהם מקורות.
4. החזר את L.

BFS(G,s):

1. for any u in V do
    a. d[u] := ∞
    b. p[u] := nil
2. Q := {s}; d(s):=0
3. while Q is not empty do
    a. u := dequeue(Q)
    b. for each v in adj(u) do
        i. if d[v] = ∞ then
            1. d[v] := d[u] + 1
            2. p[v] := u
            3. enqueue(Q,v)

DFS(G,s):

1. for all v in V
    a. k[v] := 0
    b. p[v] := nil
2. i:=0
3. while there is a vertex s with k(s)=0
    a. STACK:={s}, i:=i+1, k[s]:=i
    b. While STACK ≠ ∅
        i. u:= head(STACK)
        ii. if there is v∈adj(u) s.t. k[v]=0
            1. i:=i+1
            2. k[v]:=i
            3. p[v]:=u
            4. push(STACK,v)
        iii. else
            1. pop(STACK)

Strongly Connected Components(G):

1. Call DFS(G) to compute f[u] for all u in V
2. Compute G^T
3. Call DFS(G^T) on the vertices ordered in decreasing order of f[u] (computed in (1))
4. output the vertices in each DFS tree generated in (3) as separate component

הכלל האדום (לקשתות כבדות):
תנאי: קיים ב G מעגל C חסר קשתות אדומות, וe היא קשת לא צבועה בעלת משקל מקסימאלי (מבין הקשתות הלא צבועות) ב C.
פעולה: צבע את e באדום.

הכלל הכחול (לקשתות קלות):
תנאי: קיים ב G חתך D חסר קשתות כחולות, וe היא קשת לא צבועה בעלת משקל מינימאלי (מבין הקשתות הלא צבועות) ב D.
פעולה: צבע את e בכחול.

האלגוריתם הגנרי:
בחר קשת כלשהי עליה ניתן להפעיל את הכלל האדום או הכלל הכחול, והפעל כלל זה. עצור כאשר אין בG קשת כזו.

האלגוריתם של Prim

1. אתחול – כל הקשתות אינן צבועות; $U := \{r\}$.
2. כל עוד $U \neq V$ בצע:
   א. הפעל את הכלל הכחול על חתך $(U, V \setminus U)$ וצבע בכחול קשת $e=(u,v)$ מינימאלית בחתך זה כך ש- $u \in U, v \in V \setminus U$; בצע $U := U \cup \{v\}$.

האלגוריתם של Kruskal

1. מיין את הקשתות בסדר לא יורד לפי משקל.
2. עבור על הרשימה הממוינת, ולכל קשת $e=(u,v)$ בצע:
   א. אם יש מסלול כחול מ u ל v, צבע את e באדום,
   ב. אחרת צבע את e בכחול.

אלגוריתם לעפ"מ צהוב ביותר

נגדיר פונקציית משקל חדשה

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

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.

אלגוריתם גנרי למסלולים קלים ביותר ממקור יחיד

פונקציית חסם עליון על מרחקים (בקיצור פח"ע): זוהי פונקציה $d: V \to \mathbb{R}$ המקיימת: $d(s) \leq 0$, $d(v) \geq dist(s,v)$.

שיפור מקומי על קשת (u,v) – relaxation: בהינתן פח"ע d, שיפור מקומי על קשת (u,v) הוא הכלל: אם $d(v) > d(u) + w(u,v)$ אז $d(v) \leftarrow d(u) + w(u,v)$.

האלגוריתם הגנרי:

1. קבע פח"ע
2. כל זמן שקיימת קשת (u,v) כך ש $d(v) > d(u) + w(u,v)$ בצע שיפור (u,v).

אלגוריתם בלמן-פורד:

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

פלט: לכל צומת v - $dist(s,v) = d(v)$.

האלגוריתם:

1. אתחול: לכל $v \in V \setminus \{s\}$ אתחל $d(v) \leftarrow \infty$, $parent(v) \leftarrow nil$
   $d(s) \leftarrow 0$;
2. עבור $i = 1$ עד $V - 1$ בצע
   a. לכל קשת (u,v) בצע שיפור (u,v).

<u>אלגוריתם דייקסטרה:</u>

<u>קלט</u>: גרף ממושקל מכוון ללא משקלים שליליים, וצומת s.

<u>פלט</u>: לכל צומת v - $d(v)=dist(s,v)$

<u>האלגוריתם</u>:

1. לכל $v\in V$ אתחל $d(v)\leftarrow\infty$, ואתחל $d(s)\leftarrow 0$
2. בנוסף אתחל $Q\leftarrow V$ (Q מכיל את כל הצמתים עבורם $d(v)\neq dist(s,v)$).
3. כל זמן ש $Q\neq\emptyset$ בצע
   a. מצא ב-Q צומת u כך ש $d(u)$ מינימאלי.
   b. הוצא את u מ-Q ולכל קשת (u,v) בצע שיפור (u,v).

<u>אלגוריתם ג'ונסון:</u>

נגדיר גרף חדש $G'=(V',E')$ כך ש-$V'=V\cup\{s\}$ ו-$E'=E\cup\{sv:v\in V\}$. כמו כן נגדיר $w':E\rightarrow R$ כך ש-$w'(sv)=0$ לכל $v\in V$ ו-$w'(e)=w(e)$ לכל קשת אחרת. נסמן ב-$\delta_s(v)$ את משקל המסלול הקל ביותר מ-s ל-v ב-G'. נגדיר גם $w_{\delta_s}(uv)=\delta_s(u)+w(uv)-\delta_s(v)$.

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

<u>פלט</u>: משקל מסלול קל ביותר מכל צומת לכל צומת.

1. בנה את G' כפי שהוגדר לעיל.
2. חשב את δ_s כפי שהוגדר לעיל, בעזרת האלגוריתם של Bellman Ford.
3. חשב את פונקצית המשקל w_{δ_s}.
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 $d'(u,v)+\delta_s(v)-\delta_s(u)$.

<u>אלגוריתם חמדן למציאת מספר מקסימלי של קטעים זרים:</u>

<u>קלט</u>: אוסף משימות $S=\{a_1,...,a_n\}$. לכל משימה זמן התחלה s_i וזמן סיום f.

<u>פלט</u>: תת קבוצה מקסימלית $A\subset S$ של משימות (או קטעים) הזרים בזוגות.

<u>האלגוריתם</u>:

1. מיין את המשימות לפי סדר עולה של זמני סיום. (מעכשיו נניח כי המשימות מסודרות כך ש $f_1\leq f_2\leq\cdots\leq f_n$).
2. אתחל $A\leftarrow\emptyset$, $i\leftarrow 0$ ו- $f_0\leftarrow-\infty$. (i הוא האינדקס של המשימה האחרונה שנכללת בפתרון, ו- f_i נק' הסיום שלה).
3. עבור $m=1$ עד n בצע:
   א. אם $f_i\leq s_m$ אז $A\leftarrow A\cup\{a_m\}$ ו- $i\leftarrow m$, (אם המשימה האחרונה ב-A הסתיימה לפני ש a_m מתחילה, הוסף את a_m לA)

<u>אלגוריתם חמדן לצביעת גרף:</u>

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

<u>פלט</u>: צביעה חוקית c של G המשתמשת לכל היותר ב - d_1+1 צבעים.

1. עבור צומת צומת ע"פ הסדר הנתון של הצמתים. יהא v הצומת הנוכחי.
   א. קבע את $c(v)$ להיות הצבע המינימאלי שאינו בשימוש על ידי שכניו של v.

<u>אלגוריתם Huffman לקוד פרפיקסי אופטימלי:</u>

Recursive_Huffman($[\Sigma, f]$):

<u>קלט</u>: א"ב עם תדירות לכל אות $[\Sigma, f]$

<u>פלט</u>: עץ האפמן של $[\Sigma, f]$

1. אם $|\Sigma| = 2$ החזר עץ בעל שני עלים שהם שני איברי Σ.
2. אם $|\Sigma| > 2$:
   א. יהי $[\Sigma', f']$ הקלט הנוצר על ידי החלפת שתי אותיות x, y בעלי תדירות מינימאלית ב Σ באות חדשה z שתדירותה $f'(z) = f(x) + f(y)$.
      i. T' := Recursive_Huffman($[\Sigma', f']$).
   ב. הוסף לעלה המתאים ל- z ב- T' את x ו- y כבנים, והחזר את העץ שהתקבל T.

<u>אלגוריתם פלויד וורשל למציאת כל המסלולים הקלים ביותר:</u> (גרסה שקולה לזו שהועברה בכיתה)

שיפור מקומי באמצעות צמת ביניים (שיפור מקומי מוכלל) עבור שלשה (i,k,j) הוא הכלל:

אם $d(i,j) > d(i,k) + d(k,j)$ אז $d(i,j) \leftarrow d(i,k) + d(k,j)$

<u>קלט</u>: $G = (V, E)$ עם פונקציית משקל $w: E \rightarrow \mathbb{R}$, ללא מעגלים שליליים.

בה"כ נניח כי $V = \{1, \ldots, n\}$, ונגדיר את פונקציית המשקל המוכללת $\bar{w}: V \times V \rightarrow \mathbb{R}$ השווה ל – 0 אם $i = j$, ל - $w(i,j)$ אם $(i,j) \in E$ ול - ∞ אחרת.

<u>פלט</u>: לכל זוג צמתים $i, j \in V$ מוחזר המרחק מ- i ל- j - $dist(i,j)$.

<u>אתחול</u>: לכל זוג צמתים i, j בצע $d(i,j) \leftarrow \bar{w}(i,j)$.

עבור $k = 1$ עד n בצע

  לכל זוג צמתים i, j בצע שיפור (i,k,j).

<u>האלגוריתם קבוצה בלתי תלויה של קטעים בעלת משקל מקסימלי:</u>

<u>אתחול</u>:

1. מיין את המטלות לפי סדר עולה של זמני סיום. לאחר המיון $f_1 \leq f_2 \leq \cdots \leq f_n$.
2. אתחל $A(0) \leftarrow \varnothing$, ו- $opt(0) \leftarrow 0$.
3. לכל i מצא את $pred(i)$.

<u>גוף האלגוריתם</u>:

1. עבור $i = 1$ עד n בצע
2. אם $opt(i-1) < opt(pred(i)) + w_i$ אז
   א. $A(i) \leftarrow A(pred(i)) \cup \{a_i\}$.
   ב. $opt(i) \leftarrow opt(pred(i)) + w_i$.
3. אחרת
   א. $A(i) \leftarrow A(i-1)$.
   ב. $opt(i) \leftarrow opt(i-1)$.
4. החזר $A(n)$.

אלגוריתם תכנות דינמי לבעיית ה-Knapsack:

קלט: n פריטים $a_1, a_2, \dots, a_n$ כאשר ערכיהם $p_1, p_2, \dots, p_n$ ומשקליהם $w_1, w_2, \dots, w_n$ וכן קיבולת שק W.

פלט: הערך המקסימלי של קבוצת פריטים העומדת בקיבולת השק.

1. צור מטריצה F מגודל $(n+1)\times(W+1)$
2. אתחל $F(0,w)=0$ לכל $0 \le w \le W$
3. עבור מהשורה המתאימה ל- $k=0$ עד לשורה המתאימה ל- $k=n$
   א. חשב את כל כניסות השורה לפי הנוסחא:

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

4. החזר את $F(n,W)$

האלגוריתם הגנרי של פורד-פלקרסון לזרימת מקסימום:

1. קבע $f(e)=0$ לכל קשת e.
2. כל עוד קיים מסלול שיפור p מ s אל t ב G_f, שפר את הזרימה לאורך p.

האלגוריתם של אדמונדס-קרפ:

1. קבע $f(e)=0$ לכל קשת e.
2. כל עוד קיים מסלול שיפור מ- s אל t ב- G_f, יהא p מסלול השיפור הקצר ביותר, שפר את הזרימה לאורך p.

אלגוריתם שידוך מקסימום בגרף דו-צדדי:

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

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

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

ונצייד כל קשת עם קיבול יחידה.

קלט: גרף דו-צדדי $G=(V,E)$.

פלט: שידוך מקסימום M עבור G.

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

טענות ומשפטים שימושיים

סיווגי קשתות ביער DFS:

קשתות עץ: (u,v) קשת עץ אםם $u=p(v)$.

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

קשתות קדמיות: קשתות (u,v) מ- u לצאצא של u בעץ DFS.

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

הקשר בין סוגי הקשתות לזמני הגילוי והנסיגה הוא הקשר הבא:

קשת uv היא קשת עץ או קשת קדמית אם"ם $d(u)<d(v)<f(v)<f(u)$.

קשת uv היא קשת אחורית אם"ם $d(v)<d(u)<f(u)<f(v)$.

קשת uv היא קשת חוצה אם"ם $d(v)<f(v)<d(u)<f(u)$.

<u>משפט המסלול הלבן:</u>
ביער ה-DFS של גרף (מכוון או לא מכוון), צומת v הוא צאצא של צומת u אם"ם בזמן גילוי u (דהיינו בנקודת הזמן $d(u)$) קיים מסלול מ- u ל- v דרך צמתים שטרם התגלו.

<u>גרף הרכיבים קשירים היטב:</u>
נניח שהרק"ה ב- G הם $C_1,...,C_k$.
גרף הרכיבים קשירים היטב של G, שיסומן $G^*=(V^*,E^*)$, מוגדר באופן הבא.
$V^*=\{v_1,...,v_k\}$
$E^*=\{(v_i,v_j):$ יש ב-E קשת שיוצאת מצומת ב- C_i אל צומת ב- C_j $\}$

<u>למה:</u> גרף הרק"ה G* הוא אציקלי.
<u>למה:</u> בכל ריצת אלגוריתם DFS, כל רכיב קשיר היטב מוכל במלואו באחד מעצי ה-DFS שהאלגוריתם מחזיר.
שמורת הצבע: גרף G שחלק מקשתותיו כחולות וחלק אחר אדומות מקיים את ***שמורת הצבע*** אםם קיים בG עפ"מ שמכיל את כל הקשתות הכחולות ואף אחת מהקשתות האדומות.
<u>למה:</u> בכל ביצוע של האלגוריתם הגנרי על גרף לא מכוון משוקלל וקשיר G (שקשתותיו מלכתחילה אינן צבועות), הגרף G מקיים את שמורת הצבע.
<u>למה:</u> לכל עפ"מ קיימת ריצה של קרוסקל המוצאת אותו וריצה של פרים המוצאת אותו.
<u>למה:</u> יהא $G=(V,E)$ גרף לא מכוון וקשיר. יהיו $w:E\to R,\ w':E\to R$ שתי פונקציות משקל על הקשתות המקיימות $w(e_1)\le w(e_2)$ אם"ם $w'(e_1)\le w'(e_2)$ לכל זוג קשתות $e_1,e_2\in E$. T הוא עפ"מ לפי w_1 אם"ם T הוא עפ"מ לפי w_2.
<u>למה:</u> יהא $G=(V,E)$ גרף לא מכוון וקשיר עם פונקצית משקל $w:E\to R$. לכל משקל נתון, כל עפ"מ של G מכיל את אותו מספר של קשתות ממשקל זה.
<u>למה:</u> קשת e נמצאת באיזשהו עפ"מ של G אם"ם כל מעגל המכיל את e מכיל קשת $e'\ne e$ כך ש- $w(e')\ge w(e)$.
<u>למה:</u> קשת e נמצאת בכל עפ"מ של G אם"ם כל מעגל המכיל את e מכיל קשת e' כך ש- $w(e')>w(e)$.
<u>למה:</u> אם אין ב- $G=(V,E)$ ***מעגלים שליליים*** (שסכום משקלי קשתותיהם שלילי) שניתנים להגעה מ- s אזי לכל v שניתן להגעה מ- s קיים מסלול קל ביותר מ- s ל- v
<u>מבנה אופטימאלי של מסלולים קלים ביותר</u>: אם P הוא מסלול קל ביותר מ- u ל- v, אז כל תת-מסלול של P הוא מסלול קל ביותר בין זוג הצמתים המתאימים.
<u>אי-שוויון המשולש</u>: לכל u,v,w מתקיים $dist(u,v)\le dist(u,w)+dist(w,v)$.
<u>למה:</u> יהי $G=(V,E)$ גרף משוקלל, שאין בו מעגל בעל משקל שלילי המכיל את s. תהי $d:V\to\mathbb{R}\cup\{\infty\}$ פח"ע. אז 2 הטענות הבאות נכונות:
1. d נשארת פח"ע גם אחרי ביצוע שיפור (u,v).
2. אם לכל קשת (u,v) מתקיים $d(v)\le d(u)+w(u,v)$ (כלומר לא ניתן לבצע שיפור מקומי) אז לכל צומת v מתקיים $d(v)=dist(s,v)$.

קידוד של א"ב, קוד: נתון א"ב סופי $\Sigma=(s_1,...,s_n\}$. "קידוד של Σ" הוא מיפוי $C:\Sigma\to\{0,1\}^+$ המתאים לכל אות ב Σ מילה בינארית $C(s_i)=w_i$. קבוצת המלים $\{w_1,...,w_n\}$ היא "קוד", שגם הוא נקרא C. לכל מילה u ב Σ^+, $C(u)$ היא המלה הבינארית המתקבלת משרשור הקידודים של האותיות ב u.
קוד חד פענח: לכל זוג מלים שונות $u_1,u_2\in\Sigma^*$, צריך ש $C(u_1)\ne C(u_2)$.
קוד פרפיקסי (חסר רישות): אף מילת קוד אינה פרפיקס (רישא) של מילת קוד אחרת.
רשת זרימה: רביעיה $N=(G,s,t,c)$, כאשר $G=(V,E)$ הוא גרף מכוון ללא לולאות או קשתות מקבילות. $s,t\in V$ כאשר s הוא צומת "מקור" ו- t הוא צומת "בור". $c:E\to R^+$ היא

פונקציית קיבול המגדירה לכל קשת $e \in E$ קיבול אי שלילי $c(e)$. לכל צומת v, $in(v)$ היא קבוצת הקשתות הנכנסות לצומת ו $out(v)$ היא קבוצת הקשתות היוצאות ממנו.

פונקצית זרימה: $f : E \to R^+$, היא פונקציה המגדירה לכל קשת e את כמות הזרימה $f(e)$ העוברת בה. פונקצית זרימה חייבת לקיים שני אילוצים:

1. אילוץ הקיבול (נקרא גם חוק הקשת) לכל קשת מתקיים $0 \leq f(e) \leq c(e)$, כלומר הזרימה דרך קשת היא אי שלילית ואינה יכולה לחרוג מקיבול הקשת.
2. שימור הזרימה (נקרא גם חוק הצומת). לכל צומת $v \in V \setminus \{s,t\}$ (כלומר שאינו מקור או בור) מתקיים $\sum_{e \in out(v)} f(e) = \sum_{e \in in(v)} f(e)$, כלומר סה"כ הזרימה הנכנסת לצומת שווה לסה"כ הזרימה היוצאת מממנו.

ערך (או חוזק) פונקצית הזרימה: הזרימה נטו הנכנסת לבור. מסומן ב F או $|f|$:

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

חתך s-t ברשת זרימה:

חתך $(S,\bar{S})$ הוא חלוקה של V: $S \subset V,\ \bar{S} = V - S$. חתך *s-t* מקיים $s \in S,\ t \in \bar{S}$.

יהי $(S,\bar{S})$ חתך נתון. $(S \to \bar{S})$ היא קבוצת הקשתות החוצות את החתך מ S ל $\bar{S}$ (ב"כיוון הקדמי"): $(S \to \bar{S}) = \{(u,v) \in E : u \in S, v \in \bar{S}\}$.

$(\bar{S} \to S) = \{(v,u) \in E : u \in S, v \in \bar{S}\}$ היא קבוצת הקשתות החוצות אותו ב"כיוון האחורי".

<u>למה:</u> לכל חתך $(S,\bar{S})$ ולכל פונקצית זרימה f מתקיים

$$F = \sum_{e \in (S \to \bar{S})} f(e) - \sum_{e \in (\bar{S} \to S)} f(e)$$

קיבול של חתך $(S,\bar{S})$: סכום קיבולי הקשתות החוצות אותו בכיוון הקדמי:

$$c(S,\bar{S}) = \sum_{e \in (S \to \bar{S})} c(e)$$

<u>למה:</u> לכל חתך $(S,\bar{S})$ ולכל פונ' זרימה f מתקיים $F \leq c(S,\bar{S})$

הגרף השיורי ומסלולי שיפור:

רשת זרימה G ופונקציית זרימהf המוגדרת עליה מגדירים לכל קשת $e=(u,v)$ עם קיבול $c(e)$ וזרימה $f(e)$ שתי קשתות אנטי מקבילות עם קיבול שיורי (residual capacity):

קשת קדמית (u,v) עם קיבול שיורי $c(e)-f(e)$.

קשת אחורית (v,u) עם קיבול שיורי $f(e)$.

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

מסלול שיפור (ביחס ל-G ו-f): מסלול מ-s ל-t בגרף השיורי G_f.

<u>משפט חתך מינימום – זרימת מקסימום (Min Cut – Max Flow):</u>

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

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

<u>למה:</u> אם לכל קשת e, הקיבול $c(e)$ הוא מספר שלם אז קיימת ברשת זרימת מקסימום בשלמים, ושיטת פורד פלקרסון תמיד תעצור ותמצא זרימת מקסימום בשלמים.

זרימה במספר בורות ומספר מקורות: על מנת למצוא זרימת מקסימום כאשר ישנם מספר בורות ומספר מקורות נוסיף צומת חדש s שיהיה המקור היחיד, ונמתח ממנו קשתות בקיבול ∞ לכל אחד מהמקורות הישנים. באופן דומה, נוסיף צומת חדש t שיהיה הבור היחיד, ונמתח אליו קשתות בקיבול ∞ מכל אחד מהבורות הישנים.

זרימה עם אילוצי קיבול על הצמתים: נפצל כל צומת $v \in V$ לשני צמתים v_{in} ו-v_{out}. את כל הקשתות שנכנסו ל-v נכוון ל-v_{in} ואת כל הקשתות שיצאו מ-v נוציא מ-v_{out}. כמו כן נוסיף קשת חדשה $(v_{in} \to v_{out})$ שקיבולה $c(v)$.

סיבוכיות של אלגוריתמים

בעיה	אלגוריתם	סיבוכיות	הערות
מיון טופולוגי	מבוסס מחיקת מקורות	$O(V+E)$	DAG בלבד
מסלולים קצרים ביותר	BFS	$O(V+E)$	
	DFS	$O(V+E)$	
גרף הרכיבים קשירים היטב	מבוסס DFS	$O(V+E)$	
עץ פורש מינימום	פרים	מבוסס מערך $O(V^2)$ מבוסס ערימה $O(ElogV)$	
עץ פורש מינימום	קרוסקל	מבוסס UF $O(ElogV)$	
מסלולים קלים ביותר ממקור יחיד	בלמן פורד	$O(VE)$	מוצא מעגלים שליליים אם יש
מסלולים קלים ביותר ממקור יחיד	דיקסטרה	$O(ElogV)$	קשתות במשקל אי שלילי בלבד
מסלולים קלים ביותר בין כל זוג	ג'ונסון	$O(VElogV)$	מוצא מעגלים שליליים אם יש
מסלולים קלים ביותר בין כל זוג	פלויד-וורשל	$O(V^3)$	
מספר מקסימלי של קטעים זרים	חמדן	$O(nlogn)$	
צביעת גרף ב-d_1+1 צבעים	חמדן	$O(V+E)$	
בניית עץ Huffman	רקורסיבי	$O(nlogn)$	
קבוצה בלתי תלויה של קטעים בעלת משקל מקסימלי	מבוסס תכנות דינמי	$O(nlogn)$	
Knapsack	מבוסס תכנות דינמי	$O(nW)$	
זרימת מקסימום	כל אלגוריתם המתאים לאלגוריתם הגנרי של פורד פלקרסון	$O(Ef^*)$	f^* הוא ערך הזרימה המקסימלית ברשת. התכנסות מובטחת רק במקרה של זרימה בשלמים.
זרימת מקסימום	אדמונדס-קרפ	$O(VE^2)$	
שידוך מקסימום בגרף דו-צדדי	מבוסס זרימה	$O(VE)$	