בעיית שיבוץ משימות:

קלט: קבוצת משימות J. כל $j \in J$ מאופיין ע"י זמן התחלה $s(j)$ וזמן סיום $t(j)$. פלט: תת קבוצת $J' \subseteq J$ משימות לא חופפות בזמן, מקסימלית בגודלה.

אלגוריתם חמדן: בחירה לפי סדר סיום

סיבכויות: O(n logn) (מיון לפי סדר סיום)

בעיית מזעור האיחור:

קלט: משימה $j \in J$ מאופיינת ע"י משך ביצוע p(j) ותאריך יעד d(j) פלט: מועד התחלה s(j) לכל משימה, כך שהמשימות יתבצעו ללא חפיפה, ואחור מירבי ביחס לתאריך יעד יהיה מינימלי.

אחור מירבי: $L=\max_j L(j)$

$L(j)=\max\{0,s(j)+p(j)-d(j)\}$

שיבוץ לפי תאריכי יעד:

```
sort(J,d)
s(0) ← 0
for j ← 1 to |J| do
  s(j) = s(j-1) + p(j-1)
end for
```

סיבוכיות זמן: O(n logn)

בעיית הדפדוף:

קלט: סדרה סופית P של מספרים חיוביים (מספרי דפים מבוקשים). פלט: סדרה S של מספרים באורך זהה לאורך הקלט

אלגוריתם Belady: בכל צעד שדורש החלפה, נזרוק מהמטמון את הדף שיידרש בעתיד הרחוק ביותר (מבין כל הדפים במטמון).

לכל איבר i ב-P נחשב ערך next(i) שהוא המקום הבא בסדרה שבה מופיע דף זה:

```
for p ← 1 to #pages do
  last[p] ← 0
end for
for i ← 1 to |P| do
  next(i) ← ∞
  if last[P(i)] > 0 then
    next(last[P(i)]) ← i
  end if
  last[P(i)] ← i
end for
cache ← ∅
for i ← 1 to |P| do
  S(i) ← 0
  if P(i) ∉ cache and |cache|=k then
    S(i) ← argmax{next(p):p ∈ cache}
    cache ← cahce\{S(i)}
  end if
  cache ← cacheU{P(i)}
end for
```

סיבוכיות זמן: $O(\#pages+|P|\log k)$

k – מספר הדפים האפשרי במטמון

מציאת עץ פורש מינימום:

קלט: גרף לא מכוון וקשיר G(V,E), בייצוג של רשימת סמיכויות, משקלות חיוביים על הקשתות $w: E \to \mathbb{N}$. פלט: עץ T, תת גרף של G, מכיל את כל V ומשקל כולל של קשתותיו מינימלי.

אלגוריתם גנרי למציאת עפ"מ:

הכלל הכחול: מצא חתך שאין בו קשת כחולה, צבע את הקשת הקלה ביותר בחתך כחול.

הכלל האדום: מצא מעגל שאין בו קשת אדומה, צבע את הקשת הכבדה ביותר במעגל באדום.

האלגוריתם של Kruskal:

נמיין את הקשתות בסדר לא יורד. עבור כל קשת, אם אינה סוגרת מעגל, נוסיפה לעץ.

סיבוכיות זמן: O(|E| log|E|)

האלגוריתם של Prim:

נתחיל מ- s (קודקוד כלשהו) בכל איטרציה מוסיפים לעץ את הקשת הקלה ביותר שמגלה צומת חדש (אלגוריתם חמדן) (בעצם הקשת הקלה ביותר בחתך בין העץ לשאר הגרף).

סיבוכיות: O(|E|+|V|log|V|) ערימת פיבונצ'י.

קידוד Huffman: קידוד באורך משתנה, תווים שכיחים יותר יקודדו בפחות ביטים. ייצוג הקידו על ידי עץ בינארי: עלים יסומנו ע"י התווים; קשת לבן שמאלי תסומן ב0; קשת לבן ימני תסומן ב1. טענה: העץ המתאים לקוד אופטימלי הוא תמיד עץ בינארי מלא. סימונים: C-א"ב. אלגוריתים: סדר את המשקלים בסדר לא יורד. אחד את השניים הנמוכים ביותר 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 מימוש פשוט: $O(|C|^2)$

מימוש מתוחכם: O(|C| log|C|) (heapכ H)

מטרואידים:

הגדרה: זוג סדור (S,I) הוא מטרואיד אמא S קבוצה סופית ו-I קבוצה לא ריקה של תת קבוצות של S המקיימת:

1. אם $A \in I$ אזי לכל $B \subset A$, גם $B \in I$ (ירושה)
2. אם $A,B \in I$ ו-|A|>|B|, אזי יש $x \in A \setminus B$ כך שמתקיים $B \cup \{o\} \in I$ (החלפה)

אלגוריתם חמדן: נתונה $w: S \to \mathbb{R}$ האלגוריתם מנסה למצוא קבוצה בI בעלת משקל כולל מירבי

```
Greedy(S,w)
  A ← ∅
  while ∃x ∈ S, AU{x} ∈ I do
    x ← argmax{w(x): x ∈ S ∧ AU{x} ∈ I
    A ← AU{x}
  end while
return A
```

משפט Rado-Gale-Edmonds: תהי I קבוצה לא ריקה של תת קבוצות של S שמקיימת את תכונת הירושה: $A \subset B \in I \Rightarrow A \in I$. אלגוריתם החמדן מוצא, לכל פונקציית משקל חיובית w, קבוצה $A \in I$ בעלת משקל כולל מירבי אמם (S,I) מטרואיד.

BFS (קשירות – מסלול קצר ביותר):

סריקה לרוחב – התקדמות חזית אחרי חזית. אתחול: דרגת כל צומת, מסמן את כל השכנים של הצומת שעדיין לא סומנו, בדרגה 1 יותר מהצומת הנוכחי. יוצר עץ מסלולים הקצרים ביותר (ייתכן כי קיים יותר מעץ אחד אפשרי)

סיבוכיות זמן: O(|V|+|E|)

תכנות דינאמי:

מרחק עריכה: קלט: 2 מחרוזות x,y מעל אב Σ פלט: מספר מזערי של פעולות עריכה כדי להפוך את x ל-y. נשתמש בטבלה D.

D(i,j)=משקל מינימלי של מסלול מ(0,0) ל(i,j).

סיבוכיות: זמן O(mn), ***מקום*** O(mn) (O(m))

סכום חלקי (subset sum):

קלט: סדרה של n מספרים טבעיים a וטבעי W.

פלט: תת סדרה I עבורה $\sum_{i \in I} a_i \le W$ והסכום מקסימלי תחת אילוץ זה. נשתמש בטבלה OPT. OPT(i,w)=תת סדרה של האיברים $a_1..a_i$ שסכומםw≥ והסכום max.

בעיית מסלול קל ביותר:

קלט: גרף G(V,E), משקלות על קשתות w, צומת התחלה s וצומת סיום t. פלט: מסלול מ-s ל-t שמשקלו הכולל מינימאלי.

האלגוריתם של Dijkstra(G,w,s,t): מציאת מסלול מ-s ל-t שמשקלו הכולל מינימאלי בגרף G(V,E), ללא קשתות במשקל שלילי.
1. אתחל משקל כל צומת לאינסוף מלבד השורש שמשקלו 0. **2.** הכנס את כל הצמתים לתור עדיפויות על פי משקל המסלול הקל ביותר אליהן עד כה. **3.** הוצא את הצומת בעלת המשקל הנמוך ביותר. יהי x **4.** עדכן את שכניה: לכל שכן y – עדכן d(y) אם w(x,y) d(x)+ < d(y)

סיבוכיות זמן: בעזרת ערימת פיבונצ'י למציאת מינימום בכל איטרציה O(|E|+|V| logn|V|)

במימוש פשוט: $O(|V|^2)$

מתוך Dijkstra נגדיר עץ מסלולים קלים ביותר.

אלגוריתם Belman-Ford:

האלגוריתים מתבסס על נוסחת הרקורסיה: אם אין מעגל במשקל שלילי $d(s,x)=d_{|V|-1}[x]$ חישוב $d_i[y]$ ע"פ נוסחת רקורסיה: $d_0[s]=0$ $d_0[x]=\infty$ $\forall x \neq s$, $\forall i<0$, $\forall y \in V \rightarrow$

$d_i[y]=\min\{d_{i-1}[y],\min_{x \in V}\{d_{i-1}[x]+w(,y)\}\}$ x

האלגוריתם:

בצע |V| פעמים:

עבור על כל הקשתות בגרף בסדר כלשהו, עבור כל קשת (x,y) , אם d(x)+ w(x,y) < d(y) , עדכן את d(y) . אם באיטרציה ה - |V| היה שיפור נוסף בצומת כלשהי ← יש מעגל שלילי בגרף.

סיבוכיות זמן: O(|V||E|)

גרסה שקולה:

בצע |V| פעמים:

עבור על כל הצמתים בגרף בסדר כלשהו, עבור כל צומת x , לכל שכן y : אם d(y)+ w(y,x) < d(x) , עדכן את d(x) . אם באיטרציה ה - |V| היה שיפור נוסף בצומת כלשהי ← יש מעגל שלילי בגרף.

מסלולים קלים ביותר בין כל הזוגות:

אם אין קשתות במשקל שלילי, הרצת Dijkstra מכל צומת. ***סיבוכיות:*** $O(|V||E|+|V|^2\log|V|)$

אם יש קשתות במשקל שלילי,הרצת Bellman-Ford מכל צומת. ***סיבוכיות:*** $O(|V|^2|E|)$

Folyd-Warshall :

ממספרים את הצמתים מ1 עד |V|.

$d_k(i,j)$=משקל מזערי של מסלול מi לj שלא עובר דרך צומת שמספרו מעל k . נוסחת רקורסיה:

$d_0(i,j)=w(i,j) \quad \forall (i,j) \in E$

$d_0(i,j)=\infty \quad \forall (i,j) \notin E$

$d_k(i,j)=\min\{d_{k-1}(i,j),d_{k-1}(i,k)+d_{k-1}(k,j)\}$

אלגוריתם:

עבור $k=1..|V|$ בצע: חשב d_k על פי הנוסחה לעיל (מתבסס על מידע מאיטרציות קודמות) עבור כל זוג צמתים בגרף.

סיבוכיות זמן: $O(|V|^3)$ ***מקום:*** $O(|V|^2)$

Johnson: (לגרפים עם קשתות שליליות)

הוסף צומת s וקשת ממנה לכל צומת במשקל 0. הרץ Bellman-Ford מ-s. אם יש מעגל שלילי – התרע על כך. הגדר פונקצית משקל חדשה בגרף המקורי:

$w'(u,v)=w(u,v)+d(u)-d(v)$ הרץ Dijkstra על הגרף עם פונק' המשקל החדשה מכל צומת.

סיבוכיות: $O(VE+V^2\log V)$.

<u>סריקה לעומק –DFS:</u>

(שימוש: בדיקת קשירות, מציאת רכיבים אי-פריקים). אלגוריתם:

<u>במילים:</u> ממשיך לגלות צמתים חדשים עד אשר אין שכנים חדשים, ואז חוזר אחורה (נוסג) ומנסה לגלות שכנים חדשים בצמתים הקודמים.

סיבוכיות: $O(|V|+|E|)$

<u>**DFS** : אלגוריתם:</u>

```
DFS(G,x)
 pre[x] ← pre_c++
 for each y ∈ Adj[x] do
   if pre[y] = ∞ then
     from[y] ← x
     DFS(G,y)
   end if
 end for
 post[x] ← pre_c++
 (post[x] ← post_c++ :אם שני מונים )
end DFS
```

pre[x]=סדר הגילוי, post[x]=סדר החזרה (נסיגה) (נמנים ע"י אותו מונה, ניתן גם למנות בנפרד)

סיווג הקשתות בסריקת DFS : עבור קשת (x,y)

<u>קשת עץ:</u> אם from[y]=x (סריקתה גילתה את y)

<u>קשת קדימה:</u> pre[y]>pre[x] אבל $from[y] \neq x$

<u>קשת אחורה:</u> pre[y]>pre[x]>post[y] (סוגרת מעגל)

(במקרה של שני מונים נפרדים: pre[x]<pre[y] וגם post[y]<post[x])

<u>קשת הצידה:</u> אם השאר לא מתקיים.

הערה: בגרף לא מכוון יהיו קשתות קדימה ואחורה בלבד

הגרף T_{DFS} הנפרש ע"י קשתות העץ הוא עץ מכוון ששורשו s ופורש את כל הצמתים הנגישים מs.

<u>רכיב אי פריק</u>: בגרף לא מכוון G(V,E), יחס ~ שקילות על E שמקיים x~y אם x=y או שיש מעגל פשוט המכיל את x ו-y. תתי הגרף הנפרשים ע"י מחלקות השקילות של ~ נקראים הרכיבים האי פריקים של G. (הינם זרים בקשתות אך לא בהכרח בצמתים). $x \in V$ **<u>צומת הפרדה</u>** אםם הסרתו מגדילה את מספר הרכיבים הקשירים של הגרף. קשת $e \in E$ **<u>גשר</u>** אםם הסרתה מגדילה את מספר הרכיבים הקשירים של הגרף.

bc(G) = הגרף שצמתיו הם : הרכיבים האי-פריקים של G וצמתי ההפרדה של G. קשתותיו הן: מחברות בין כל הזוגות של רכיב אי-פריק וצומת הפרדה שמוכל בו. גרף זה הינו עץ.

אלגוריתם למציאת רכיבים אי פריקים:

לכל צומת נשמור pre[x] (זמן גילוי ב DFS) ו – low[x] (pre של הצומת הגבוהה ביותר בעץ DFS אליה ניתן להגיע בתנועה על תת העץ של x והשימוש בקשת אחורה אחת). מריצים DFS, בכל צומת שמגלים נעדכן pre[x] בעת הגילוי ונאתחל: low[x] = pre[x] . אם ניתקל בקשת אחורה נעדכן low[x] אם קטן יותר מהקיים. בעת הנסיגה לאב y נעדכן אם יש צורך את low[y] לזה של הצאצא. אם לא נעשה עדכון באב אזי האב הוא צומת הפרדה (מלבד המקרה שהאב הוא השורש) . השורש יהיה צומת הפרדה רק אם בריצת DFS יהיו לו שני בנים ומעלה.

סיבוכיות: זמן $O(|V|+|E|)$ ***מקום*** $O(|V|+|E|)$

<u>חלוקת הגרף לרכיבים קשירים היטב:</u> בכל רכיב יש מסלול מכוון מא לץ וגם מץ לא.

<u>אלגוריתם למציאת רכיבים קשירים היטב:</u>

1. בצע DFS-ים עד אשר כל הצמתים נסרקו.
2. מיין הצמתים בסדר יורד על ערך ה post.
3. עבור על כל הצמתים על פי הסדר הממוין:
   3.1 אם עדיין לא סומן כשייך לרכיב קשיר היטב:
   3.2 החל בצומת זה סריקת DFS על גרף ההיפוך (הפיכת כיוון הקשתות).
   3.3 כל צומת שתצליח להגיע אליו בסריקה החדשה שייכת לאותו רכיב קשיר היטב
4. עבור לצומת הבא שעדיין לא סומן.

סיבוכיות: $O(|V|+|E|)$

<u>זרימה ברשתות:</u>

רשת זרימה = גרף מכוון G(V,E) עם פונקציית קיבול $c:E \to \mathbb{N}$, מקור s ובור t.

זרימה ברשת G,c,s,t, פונקציה f על הקשתות:

1. $\forall e \in E,\ 0 \le f(e) \le c(e)$
2. $\forall v \in V \setminus \{s,t\},\ \sum_{e \in in(v)} f(e) = \sum_{e \in out(v)} f(e)$

ערך $|f| = \sum_{e \in out(s)} f(e) - \sum_{e \in in(s)} f(e)$

<u>חתך</u> הוא חלוקה של V ל2 קבוצות לא ריקות S ו- V\S , כש-S ∈ s ו-V\S ∈ t. סימון (S,V\S)

<u>קיבול</u> החתך(S,V\S) הוא סך כל הקיבולים על הקשתות החוצות את החתך מ – S ל – V/S

<u>חתך מינימום</u> הוא חתך *s-t* אשר קיבולו קטן-שווה לקיבול כל חתך אחר.

<u>קיבול שיורי</u> של זוג צמתים u,v מייצג את כמות הזרימה שניתן להוסיף ולדחוף מ- u ל- v: $c_f(u,v) = c(u,v) - f(u,v)$.

<u>הגרף השיורי:</u> $G_f(V,E_f)$ הוא הגרף המתקבל ע"י הקשתות $(u,v) \in V \times V$ כך ש:

$$c_f(u,v) > 0$$

<u>מסלול משפר</u> הינו מסלול מ s ל t בגרף השיורי

<u>משפט Min-Cut-Max-Flow</u>: הטענות שקולות:

1. f פונקצית זרימה מקסימלית.
2. אין מסלול שיפור מ-s ל-t בגרף השיורי.
3. קיים חתך s-t שקיבולו |f| - חתך מינימאלי.

אלגוריתם Ford-Fulkerson:

1. מצא מסלול מ-s ל-t כלשהו.
2. הרווה אותו (העבר בו זרימה גדולה ככל האפשר)
3. בנה רשת שיורית על פי זרימה זו.
4. בצע שוב על הרשת השיורית (מצא מסלול משפר), עד שאין יותר מסלולי שיפור.

סיבוכיות: $O(f^*(|V|+|E|))$ כאשר f^* הינה הזרימה המקסימלית.

שיטת הסילום – Scaling וריאציה על Ford-Fulkerson: נאתחלן delta להיות החזקה הגדולה ביותר של 2 שאינה גדולה מהקיבול המירבי של קשת היוצאת מ – s.

דומה ל Ford-Fulkerson אך כעת נדרוש כי קיבול המסלול המשפר יהיה לפחות delta . אם לא קיים נעדכן: delta ← delta/2 . ונבצע שוב.

סיבוכיות: $O(|E|2 \cdot \log C)$ כאשר C הינו הערך אותו אתחלנו להיות delta.

אלגוריתם Edmonds-Karp:

וריאציה על Ford-Fulkerson . השינוי: בשלב 1 נשתמש ב BFS – ונבחר את המסלול הקצר ביותר.

סיבוכיות: $O(|V||E|^2)$

אלגוריתם Dinic:

<u>הגדרה:</u> זרימה f ברשת G,c,s,t הינה זרימה חוסמת אםם כל מסלול מs לt כולל לפחות קשת אחת עבורה ערך הזרימה שווה לקיבול.

<u>הגדרה:</u> עבור רשת זרימה G,c,s,t תת הרשת L(G),c,s,t שמתקבלת מ G,c,s,t ע"י הסרת כל הקשתות שאינן נמצאות על מסלול קצר ביותר מ s ל t נקראת רשת השכבות של G,c,s,t

<u>הרעיון:</u> בכל איטרציה נמצא זרימה חוסמת g ברשת השכבות ונעדכן f ← f+g

סיבוכיות: $O(|V|^2|E|)$

<u>מציאת שידוך מקסימאלי/מושלם:</u>

שידוך בין קבוצה X לקבוצה Y.

רדוקציה לבעיית זרימה:

V= X U Y U {s,t} ,

$$E = \{(s,x) \mid x \in X\} \cup \{(y,t) \mid y \in Y\} \cup \{(x,y) \mid x \in X, y \in Y\}$$

$$c(s,x) = c(y,t) = 1, c(x,y) = \infty$$

גודל השידוך המקסימאלי כערך זרימת המקסימום

סיבוכיות: (בעזרת אלג' דיניץ') $O\left(\sqrt{|V|} \cdot |E|\right)$

משפט Hall: בגרף דו צדדי קיים שידוך מלא אמ"מ לכל תת קבוצה ב-X יש מספר שכנים גדול או שווה למספר הצמתים בתת הקבוצה.

<u>בעיית הנישואים היציבים:</u>

חוסר יציבות: אם א' נשואה לב' וג' נשואה לד' ומתקיים אחד המצבים הבאים:

1.א' מעדיפה את ד' וד' מעדיף את א'.

2.ב' מעדיף את ג' וג' מעדיפה את ב'.

אלגוריתם Gale-Shapley:

צד אחד (נשים, בהג"כ) יציע והצד השני יסכים או ידחה על פי הקריטריונים להלן.

- בוחרים אישה (x) שאינה משודכת ועדיין לא ניסתה את כל הגברים.
- x מציעה ל-y שהוא הראשון ברשימה שלה שעדיין לא ניסתה.
- אם y לא משודך אזי x ו- y משתדכים.
- אם y משודך , אבל מעדיף את ההצעה החדשה אזי x ו- y משתדכים.
- הפלט הוא אוסף הזוגות המשודכים בסוף.

סיבוכיות: $O(n^2)$

פלט האלגו' הוא שידוך מושלם ומהווה נישואים יציבים.

<u>הגדרות:</u> עבור $x \in X$ נאמר ש$y \in Y$ הוא **בן זוג אפשרי** אםם יש פתרון כלשהו לבעיית הנישואים היציבים הכולל את (x,y).

best(x) – בן הזוג האפשרי העדיף ביותר לx

worst(x) – בת הזוג האפשרית עם העדיפות הנמוכה ביותר עבור y

<u>משפט:</u> אלגו' Gale-Shapley תמיד משדך לכל אישה $x \in X$ את best(x) ולכל גבר $y \in Y$ את worst(y) (או להיפך, אם הגברים עשו את הבחירה).

<u>שיטת "הפרד ומשול" :</u>

<u>קונבולוציה:</u> זהו הווקטור c עבור $c_i = \sum_{j=0}^{i} a_j b_{i-j}$

שני פולינומים p,q שמקדמיהם הם הווקטורים a,b חישוב קונוולוציה יתבצע ע"י:"

כפל איבר איבר $a*b = FFT^{-1}(FFT(a) \bullet FFT(b))$

סיבוכיות זמן FFT: $T(n) = 2T(n/2) + O(n)$

$T(n) = O(n \log n)$

<u>מרחק עריכה:</u> ניתן לחשב מרחק עריכה ב- O(mn) כאשר סיבוכיות המקום O(min{m,n})