BFS- גרף G לא מכוון וצומת התחלתי s. מוצא לכל צומת בר השגה מ-s את המרחק הקצר ביותר בקשתות. מייצר כפלט עץ מסלולים קצרים ביותר של הצמתים הישיגים מ-s .

$\lambda(v)$ - המרחק המינימאלי של v מ-s.

pred[v] - הצומת שגרם ל-v להתגלות. אב של v.

זמן ריצה- O(V+E).

*לכל e=(u,v) $dist(s,v) \le dist(s,u)+1$.

האלגוריתם:

1. לכל $v \in V \setminus \{s\}$, $\lambda(v) \leftarrow \infty$, $pred(v) \leftarrow null$.
2. עבור s: $\lambda(s) \leftarrow 0$, $pred(s) \leftarrow null$.
3. הכנס את s לתור Q וכל עוד $Q \ne \phi$:

3.1 הוצא את הצומת בראש התור Q – u.

3.2 לכל שכן v של u עבור $\lambda(v)=\infty$:

3.2.1 $\lambda(v)=\lambda(u)+1$.

3.2.2 $pred(v) \leftarrow u$.

3.2.3 הכנס את v לסוף התור Q.

DFS- גרף מכוון וצומת התחלתי s. מחזיר כפלט מספור של הצמתים ויער DFS המורכב מקשתות $(p(u),u)$ או עץ יחיד אם כל הצמתים ישיגים מ-s – היער נקרא G_π .

k[v]- זמן הגילוי של הצומת v.

p[v] - הצומת שגרם ל-v להתגלות. אב של v.

f[v]- זמן סיום הטיפול בצומת v (זמן הנסיגה).

קשת עץ-(u,v),אם u התגלה ע"י סריקת (u,v).

קשת אחורית- (u,v) אם מחברת את u לאביו הקדמון v בעץ DFS. לולאה עצמית נחשבת כקשת אחורית.

קשת קדמית-(u,v) מחברת את u לצאצא שלו v בעץ DFS ואינה קשת עץ.

קשת חוצה- קשת המחברת בין 2 צמתים ללא קשר אב-קדמון – צאצא.

סיבוכיות זמן- O(V+E).

משפט – נניח ש $u_1,...,u_k$ מתגלים אחרי שמרכז הפעילות הגיע ל-v ולפני הנסיגה. אז הם צאצאים שלו.

משפט המסלול הלבן- ביער DFS מכוון/לא מכוון צומת v הוא צאצא של u אם"ם כש-u התגלה קיים מסלול מ-u ל-v שמכיל רק צמתים שעוד לא התגלו.

- בגרף לא מכוון לכל שני צמתים שמחברת ביניהם קשת יש קשר של אב קדמון – צאצא.
- בגרף לא מכוון יש רק קשתות עץ וקשתות אחוריות.

האלגוריתם:

1. לכל $u \in V$, $k(u)=0$, $p(u) \leftarrow null$.
2. לכל $e \in E$, $e='new'$, $i=1$.
3. כל עוד קיים צומת חדש u ($k(u)=0$) בצע:

3.1 $v \leftarrow u$, $k(v) \leftarrow i$

3.2 $i=i+1$

4. כל עוד יש ל-v קשת חדשה (new) או p(v) מוגדר בצע:

4.1 אם יש ל-v קשת חדשה $e=v\to w$ סמן $e='old'$

4.1.1 אם $k(w)=0$ בצע:

4.1.1.1 $p(w) \leftarrow v$, $k(w) \leftarrow i$.

4.1.1.2 $v \leftarrow w$, $i \leftarrow i+1$.

4.2 אחרת (p(v) מוגדר) בצע $v \leftarrow p(v)$.

FFT: נתונים 2 וקטורים $(a_0,a_1,...,a_{n-1})$ ו-$(b_0,b_1,...,b_{n-1})$ ונחשב $a*b=(c_0,c_1,...,a_{2n-2})$ כאשר $c_k = \sum_{i,j<n,i+j=k} a_i b_j$.

האלגוריתם: סיבוכיות O(nlog(n)).

1) בהינתן וקטורי המקדמים נחשב את הערכים שלהם ב-2n נקודות (שורשי היחידה).

2)נבצע לכל $1\le j\le 2n$ $C(x_j)=A(x_j)B(x_j)$.

3)נחשב את מקדמי C ע"פ ערכיו ב2n נקודות.

שלב 3- נגדיר $D(x)=\sum_{s=0}^{2n-1} d_s x^s$ כאשר $d_s=C(w_{s,2n})$ השורש ה-2n שמספרו s. אזי $c_s=\frac{1}{2n}D(w_{2n-1-s,2n})$

מיון טופולוגי: עבור גרפים מכוונים ללא מעגלים – DAG. נותן סידור של הצמתים כך לכל $u,v\in V$ כך שקיימת קשת (u,v), u יופיע לפני v בסידור.

סיבוכיות זמן: O(V+E).

למה- G מכוון הוא חסר מעגלים אם"ם בכל הרצת DFS על G אין קשתות אחוריות.

רכיבים קשירים היטב: G גרף מכוון. רכיב קשיר היטב הוא תת קבוצה מקסימאלית של צמתים כך שלכל u,v בקבוצה יש מסלול מכוון מ-u ל-v וגם להפך.

G^t-G עם קשתות הפוכות. רכיבי הקשירות היטב בהם זהים. ניתן ליצרו ב-O(V+E) אם G נתון ברשימת פגיעות.

סיבוכיות זמן: O(V+E)

אלגו: -מריצים DFS לחישוב f[u].

- בונים את G^t ומריצים DFS עליו אך בלולאה הראשית בוחרים צמתים לפי סדר יורד של f[u].
- מחזירים צמתי כל עץ ביער כרכיב.

גרף הרכיבים G^*- רכיבי הקשירות הם הצמתים ויש קשת בין שניים שונים אם יש קשת מצומת ברכיב אחד לשני. זהו גרף מכוון וחסר מעגלים מכוונים (DAG).

טענה: U רכיב קשיר היטב. אם $u,v\in U$ אז כל מסלול ביניהם מכיל רק צמתים מ-U.

הקדמון של u- בריצת DFS נתונה הוא צומת w שקיים מסלול מ-u אליו והוא בעל זמן הנסיגה הגדול מבין הצמתים שיש מסלול מ-u אליהם. יסומן ב- $\phi(u)$.

טענה: לכל ריצת DFS u ו- $\phi(u)$ באותו רכיב קשיר.

טענה: לכל u,v ולכל ריצת DFS u ו-v באותו רכיב קשיר היטב אם"ם $\phi(u)=\phi(v)$ בריצת הDFS.

טענה: לכל ריצת DFS $\phi(u)$ הוא אב קדמון של u בעץ

צמתי הפרדה ורכיבים אי פריקים-עבור גרף קשיר.

צומת הפרדה- v צומת הפרדה אם קיימים $a,b\in V$ שונים מ-v כך שכל מסלול ביניהם עובר דרך v.

שקול -הסרתו הופכת את הגרף ללא קשיר.

רכיב אי-פריק- תת קבוצה (מקסימאלית) של קבוצת הצמתים אשר הגרף המושרה לא מכיל צמתי הפרדה ותת הקבוצה מכילה לפחות צומת הפרדה אחד של הגרף

מבנה העל $\tilde{G}$ - גרף בו הצמתים הם רכיבים אי פריקים וצמתי הפרדה ויש קשת בין כל צומת הפרדה לרכיב אי פריק המכיל אותו. לכל גרף G, $\tilde{G}$ הוא עץ.

L[v]- הערך k[u] המינימאלי עבור u שניתן להגיע מ-v אליו ע"י קשתות עץ ולכל היותר אחורית אחת.

טענה-צומת פנימי u הוא צומת הפרדה אם"ם קיימת קשת עץ בעץ DFS (u,v) כך ש $k[u]\le L[v]$.

טענה-השורש של עץ DFS הוא צומת הפרדה אם"ם יש לו לפחות שני בנים.

האלגוריתם בסיבוכיות: O(V+E)

עפ"מ: גרף קשיר ולא מכוון. יש משקולות על הקשתות. המטרה למצוא עץ פורש שסכום המשקולות בו מינימאלי

הכלל האדום: בחר מעגל ללא קשתות אדומות וצבע באדום את הקשת הלא צבועה בעלת משקל מקסימאלי.

הכלל הכחול: בחר חתך שלא מכיל קשתות כחולות וצבע בכחול את הקשת עם המשקל המינימאלי.

האלגוריתם הגנרי: בכל שלב הפעל את הכלל הכחול או האדום עד שכל הגרף צבוע. הקשתות הכחולות מרכיבות עפ"מ.

משפט: נניח בהינתן עץ T שלכל $e\in T$ מתקיים לכל $e'\in C(e)$ ($C(e)$-החתך שמגדירה e בT) $w(e)\le w(e')$. אז T עפ"מ.

האלגוריתם של Prim: O(E+VlogV)

-הגדר עץ T והכנס אליו צומת שרירותית s.

-עבור על כל הקשתות היוצאות מצומת ב-T לצומת ב-V\T ובחר את הקשת הקלה ביותר.

-הכנס את הקצה השני שלה לT.

-צבע כל קשת פנימית בעץ באדום.

-ממשיכים עד שT מכיל את כל צמתי V.

האלגוריתם של Kruskal: O(ElogV)

-מיין את הקשתות לפי משקלן בסדר לא יורד.

-נעבור על הקשתות ממשקלן נמוך לגבוה ואם הקשת סוגרת מעגל כחול בעץ הקיים אז לא נוסיף אותה (נצבע באדום), אחרת נוסיף אותה (נצבע בכחול).

משפט: נניח שלכל $e\in G$, $e\notin T$, הקשת e היא הכבדה ביותר במעגל שנוצר ע"י $T+\{e\}$. אזי T עפ"מ

מסלולים קלים ביותר: גרפים מכוונים או לא מכוונים. לכל קשת יש משקל ומשקלו של מסלול הוא סכום המשקלות על הקשתות.

מסלולים קלים ביותר ממקור יחיד s- מניחים שאין מעגלים שליליים.

אי שוויון המשולש: נתון מקור s וקשת $u\to v$. אזי $\delta(s,v)\le\delta(s,u)+w(u\to v)$.

האלגוריתם הגנרי:

-לכל $v\in V$, $d[v]\leftarrow\infty$, $d[s]\leftarrow 0$.

-כל עוד קיימת קשת e=(u,v) כך ש-d[v]>d[u]+w(e) אז d[v]=d[u]+w(e).

האלגוריתם של בלמן-פורד O(VE):

- לכל $v\in V$, $d[v]\leftarrow\infty$, $\pi(v)=null$
- $d[s]\leftarrow 0$

-לכל $1\le i\le |V|-1$ עבור על כל הקשתות (u,v) ובצע:

-אם $d[v]>d[u]+w(u\to v)$ אז:

- $d[v]\leftarrow d[u]+w(u\to v)$
- $\pi(v)\leftarrow u$

-עוברים שוב על כל הקשתות בגרף ואם יש $u\to v$ כך ש $d[v]>d[u]+w(u\to v)$ אז פולטים "יש מעגלים שליליים".

האלגוריתם של דיקסטרה:

קלט: גרף מכוון ממושקל עם משקלים אי שליליים.

פלט: לכל צומת v , dist(s,v)=d(v).

האלגוריתם:

-לכל $v\in V$, $d[v]\leftarrow\infty$, $d[s]\leftarrow 0$.

-כל הצמתים ב-V מוכנסים לתור Q.

-כל עוד $Q\ne\phi$: הוצא מQ צומת עם d[u] מינימאלי.

-לכל קשת (u,v) בצע שיפור (u,v).

סיבוכיות זמן: O(E+VlogV)/ O(ElogV).

*באלגוריתם של דיקסטרה אם הגרף לא מכוון ניתן לפצל כל קשת לשתי קשתות אנטי-מקבילות ולהריץ.

אלגוריתם לבניית עץ מסלולים קלים ביותר:

-בנה תת גרף המורכב מכל הצמתים ב-G ומאיחוד הקשתות על המסלולים הקלים ביותר לכל צומת.

-הרץ BFS על תת הגרף מצומת המקור.

-החזר את עץ ה-BFS (הצמתים והקשתות ששימשו לגילוי צמתים בהרצת הBFS).

האלגוריתם של Johnson:

קלט: גרף מכוון

פלט: אם יש מעגל שלילי בגרף מכריז ועוצר. אחרת לכל שני צמתים u,v מחזיר את משקל המסלול הקל ביותר מ-u ל-v שהוא d(u,v).

האלגוריתם:

-הוסף צומת s וקשת ממנו במשקל 0 לכל צומת.

-הרץ Belman-Ford מ-s וודא שאין מעגל שלילי.

-לכל קשת הגדר פונקצית משקל חדשה:

w'(u,v)=w(u,v)+d(u)-d(v)

-הרץ דיקסטרא מכל צומת בגרף.

-תקן לכל צומת את משקל המסלול הקל:

d(u,v)=d'(u,v)-d(u)+d(v)

סיבוכיות זמן – O(V(E+VlogV))

השלמת זרימה:

- הגרעין של G- S(G) כל הקשתות שנמצאות על מסלול קצר ביותר מ-s ל-t.

טענה: 'G ו-"G שתי רשתות שיוריות באיטרציות עוקבות של EK. אז לפחות אחד מהשניים קורה:

1. $\delta_{G'}(s,t)<\delta_{G''}(s,t)$
2. $|S(G')|>|S(G'')|$

אלגוריתם חמדן: צביעה חוקית של גרפים

האלגוריתם: עבור סידור נתון של הצמתים בגרף האלגוריתם יעבור צומת צומת ויתן לכל צומת את הצבע המינימאלי במספרו מרשימת הצבעים.

סיבוכיות זמן : O(V+E).

-האלגוריתם ייתן צביעה אופטימאלית אם ערך צביעה זו הוא הדרגה המקסימאלית של צומת בגרף פלוס 1.

-טענה: לכל גרף קיים סידור של הצמתים כך שהאלגוריתם יחזיר ערך צביעה אופטימאלי.

אלגוריתם חמדן: צביעת גרף אינטרוולים

-מיין את הצמתים לפי זמן ההתחלה של האינטרוולים si והרץ את האלגוריתם לצביעה על הסידור הזה.

סיבוכיות זמן : O(|V|log|V|+|E|).

עצי האפמן וקודים פרפיקסים אופטימאליים :

האפמן בינארי:

-רוצים לבחור קידוד לכל תו כך שנקבל ייצוג של הנתונים בקובץ במספר מינימאלי של ביטים.

-ייצוג ע"י קוד פרפיקס- ע"י עץ כאשר האותיות הן העלים. 0 אומר לפנות לבן השמאלי ו-1 לימני. כל צומת בעץ מכיל את סכום התדירויות של עלים בתת עץ שלו.

- תדירות = בעצם הסתברות המופע של תו בקובץ.

סימונים: - C- אוסף התווים בקובץ.

- $d_T(c)$ - עומק המסלול של התו c בעץ T או אורך המחרוזת של קידוד התו.

-f(c)- תדירות (או הסתברות) ההופעה של c בקובץ.

- $B(T)=\sum_{c\in C} d_T(c)f(c)$ - מחיר העץ T

-קוד אופטימאלי = B(T) מינימאלי. יתקבל עבור עץ מלא. מספר העלים |C| ומספר הצמתים הפנימיים |C|-1.

למה: קיים עץ T אופטימאלי בו שני התווים בעלי התדירות המינימאלית מופיעים כעלים אחים עמוקים ביותר. **סיבוכיות זמן האפמן רקורסיבי** O(|C|log(|C|))

האפמן סיגמא-ארי:

-בעץ סיגמא-ארי מלא עם k צמתים פנימיים יש $n=(\sigma-1)k+1$ עלים.

-T עץ סיגמא-ארי אופטימאלי ו-L גובה העץ, אז כל הצמתים הפנימיים בגובה קטן מ-L-1 מלאים.

טענה: קיים עץ T סיגמא-ארי אופטימאלי עם גובה L ש

1.כל הצמתים הפנימיים בעומק L-1 מלאים חוץ מאולי צומת פנימי אחד.

2.מספר הבנים של צומת זה σ' ולהם משקל מינימאלי מכל העלים. $\sigma'=(n-2)\bmod(\sigma-1)+2$.

האלגוריתם עבור א"ב סיגמא-ארי Huffman(W):

1. אם |W|=1 החזר $T=(W,\phi)$.
2. $\sigma'=(n-2)\bmod(\sigma-1)+2$
3. בחר Wmin קבוצה של σ' איברים מינימאליים בW.
4. צור מילה חדשה w'.הגדר $f(w')=\sum_{w\in W_{min}} f(w)$
5. נגדיר $f(w')=\sum_{w\in W_{min}} f(w)$ ונבצע T=Huff(W')
6. הפוך את w' לצומת פנימי ע"י הוספת אברי w' ל-T כילדים של w' והחזר את T.

מסלולים קלים ביותר בין כל זוג צמתים (דינאמי):

1) $d_{i,j}^{(k)}$ - משקל המסלול הקל ביותר מ-i ל-j כאשר צמתי הביניים במסלול שייכים ל- $\{1,2,...,k\}$.

המטרה: לחשב את $d_{i,j}^{(|V|)}$.

אתחול לכל i,j:
$$d_{i,j}^{(0)}=\begin{cases}0 & i=j\\ w(i,j) & (i,j)\in E\\ \infty & \text{אחרת}\end{cases}$$

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

האלגוריתם בסיבוכיות: $O(|V|^3)$.

2) $d_{i,j}^{(m)}$-משקל המסלול הקל ביותר שמשתמש בלכל היותר m קשתות. המטרה: לחשב את $d_{i,j}^{(|V|-1)}$.

אתחול לכל i,j :
$$d_{i,j}^{(1)}=\begin{cases}w(i,j) & (i,j)\in E\\ \infty & \text{אחרת}\end{cases}$$

$$d_{i,j}^{(m)}=\min_{1\le k\le |V|}\{d_{i,k}^{(m-1)}+w(k,j)\}$$

הנחה: לשני האלגו' $w(i,i)=0$ לכל i.

האלגוריתם בסיבוכיות זמן: $O(|V|^4)$.

תכנות דינאמי-קבוצה בת"ל של אינטרוולים עם משקל מקסימאלי:

יש לנו אוסף של אינטרוולים a1,...,an עם זמן התחלה si, זמן סיום fi ורווח wi>=0. אנו רוצים למצוא תת קבוצה של משימות שתהיה בת"ל ורווחה מקסימאלי.

Opt(i)- המחיר האופטימאלי עבור המשימות a1,…,ai.

-נניח כי הקלט ממויין בסדר לא יורד של fi. כלומר מניחים כי $f_1\le f_2\le..\le f_n$.

Pred(j)- האינדקס של הקטע שמסתיים הכי מאוחר מבין כל הקטעים שמסתיימים לא אחרי ש- i מתחיל.

$$opt(i)=\max\begin{cases}opt(i-1)\\ opt(pred(i))+w(i)\end{cases}$$

$$opt(0)=0$$

סיבוכיות זמן האלגוריתם : O(n) בהנחה שהאינטרוולים ממויינים וpred(i) נתון לכל אינטרוול.

תכנות דינאמי-סידור אופטימאלי לכפל n מטריצות

נתונה שרשרת של מטריצות $\{A_1,...A_n\}$ כאשר למטריצה A_i מימדים $p_{i-1}\times p_i$. אנו רוצים לדעת איך לסדר את הסוגריים כך שהכפלת המטריצות תגרור כמות הכפלות מינימאלית.

- הכפלת מטריצה מסדר pxq עם מטריצה מסדר qxr דורשת ביצוע pqr מכפלות.

-m[i,j] מס ההכפלות המינימאלי עבור $A_i\times..\times A_j$.

$$m[i,j]=\begin{cases}0 & i=j\\ \min_{i\le k\le j-1}\{m[i,k]+m[k+1,j]+P_{i-1}P_kP_j\}\end{cases}$$

-נמלא את המטריצה לפי האלכסונים.

-m[1,n] זה מספר ההכפלות המינימאלי שאנו רוצים.

סיבוכיות זמן – $O(n^3)$.

תכנות דינאמי-Sequence Alignment

-נתונות שתי מחרוזות $x=x_1x_2....x_m$, $y=y_1y_2....y_m$.

-שידוך M בין מחרוזות מקיים:

1. שידוך בין $\{1,...,m\}$ ל- $\{1,...,n\}$.
2. אם $(i,j),(i',j')\in M$ וגם $i<j\Leftarrow i'<j'$ (אין הצטלבויות בשידוך).

-עלות שידוך M:

1. כל מיקום שאינו משודך עולה $\delta>0$.
2. $(i,j)\in M$, נשלם עליה $\alpha(x_i,x_j)$ כשבדרך כלל $\alpha(b,b)=0$.

-המרחק בין x ל-y - עלות השידוך הזול ביותר.

טענה: אם $(m,n)\notin M$, או ש-m ב-X לא שודך, או ש-n ב-Y לא שודך או שניהם.

הנוסחה הרקורסיבית:

$$Opt(i,j)=\min\{\alpha(x_i,x_j)+Opt(i-1,j-1),\ \delta+Opt(i-1,j),\ \delta+Opt(i,j-1)\}$$

תנאי התחלה: $Opt(i,0)=i\delta$, $Opt(0,j)=j\delta$.

ממלאים את המטריצה שורה שורה.

סיבוכיות זמן האלגוריתם : O(mn)

רשת זרימה: זהו גרף מכוון, בו לכל קשת קיבול אי-שלילי ונתונים שני צמתים s ו-t.

זרימה חוקית צריכה לקיים:

-חוק הקשת- לכל קשת e מתקיים $0\le f(e)\le c(e)$.

-אילוצי שימור- לכל צומת $v\in V\setminus\{s,t\}$ סכום הזרימה הנכנסת לצומת שווה לסכום הזרימה היוצאת.

-ערך זרימת מקסימום- $|f|=\sum_{e\in\delta^+(s)}f(e)-\sum_{e\in\delta^-(s)}f(e)$

- $\delta^+(v)$ - אוסף הקשתות היוצאות מצומת v ב-G.

קיבול של חתך- סכום קיבולי הקשתות שחוצות את החתך מכיוון s ל-t.

זרימה על חתך- סכום הזרימה על הקשתות החוצות את החתך מכיוון s ל-t פחות סכום הזרימה על הקשתות החוצות את החתך מכיוון t ל-s.

גרף שיורי- לכל קשת e עם קיבול c(e) יש שתי קשתות:

-קשת קדמית- קיבול c(e)-f(e) בכיוון הקשת המקורית.

-קשת אחורית- f(e) הפוכה בכיוונה מהקשת המקורית.

ביצוע השיפור- Δ - הקשת במסלול שיפור שקיבולה השיורי מינימאלי. נגדיל את הזרימה של כל קשת המתאימה לקדמית ב- Δ ונוריד ב- Δ את הזרימה לכל קשת שמתאימה לאחורית. יגדיל את הזרימה ב- Δ .

Min cut-Max flow_ הבאים שקולים:

- f זרימת מקסימום
- אין מסלול שיפור מ-s ל-t בגרף השיורי.

-קיים חתך s-t שעבורו |f|=C(S,V\S). זהו חתך מינימום בו כל הקשתות רוויות.

אלגוריתם Ford-Fulkerson:

-לכל $e\in E$, f(e)=0.

-כל עוד קיים מסלול שיפור בגרף השיורי, בצע שיפור.

סיבוכיות: O(f*E).

-אם הקיבולים שלמים האלגו' מוצא זרימה בשלמים.

אלגוריתם אדמונדס-קרפ:

-בדיוק כמו FF אך בכל שלב נבחר מסלול השיפור כמסלול הקצר ביותר מ-s ל-t (BFS).

סיבוכיות: $O(E^2V)$.

מציאת זרימת מקס' כשיש מס' מקורות ומס' בורות:

-מוסיפים צומת s וקשת ממנו לכל מקור עם קיבול ∞ .

-מוסיפים צומת t וקשת מכל בור אליו עם קיבול ∞ .

-אם יש אילוץ קיבול על הצמתים נפצל צומת לשני צמתים ובאמצע נעביר קשת בודדה עם קיבול הצומת.

שידוך מקסימום בגרף לא מכוון (דו"צ):

שידוך- תת קבוצה של קבוצת הקשתות כך שלכל שתי קשתות אין נקודת קצה משותפת.

שידוך מלא- מכסה את קבוצת הצמתים הקטנה יותר.

שידוך מושלם- מכסה את כל צמתי הגרף.

משפט: ב-G' יש זרימה מעוצמה k אמם ב-G יש שידוך בגודל k.

סיבוכיות זמן: O(VE).

הגדרה: $\Gamma(A)=\{v\in R|\exists(u)(u\in A,(u,v)\in E\}$, כאשר $A\subseteq L$. זו קבוצת הצמתים מ-R שהם השכנים של צמתים מ-A.

משפט Hall: בגרף דו"צ $G=(L\cup R,E)$ יש שידוך מלא בו כל צומת מ-L משודך (מושלם אם |L|=|R|) אמם לכל $A\subseteq L$ מתקיים $|\Gamma(A)|\ge|A|$.

משפט חשוב: זרימת מקסימום שווה ל-k אמם מספר המסלולים זרים בקשתות בין s ל-t הם k.

אלגוריתם חמדן: שיבוץ משימות – מזעור האיחור:

הרצאה: ישנן אוסף מטלות a1,a2,…,an ולכל מטלה זמן ריצה ti וזמן סיום מקסימאלי di.

איחור של משימה a_i : $l_i=\max\{0,f_i-d_i\}$

כאשר si ו-fi הם הזמנים שניתנו ל-ai בשיבוץ הנתון.

המטרה: למצוא שיבוץ (מתן זמני התחלה וסיום לכל משימה שממזער את השיבוץ המקסימאלי:

$\min_{\text{שיבוצים}}\{\max_{\text{משימות}}\{l_i\}\}$.

האלגוריתם סיבוכיות זמן : O(nlog(n)):

1. מיין את המשימות ע"פ זמני הסיום di.
2. שבץ את המשימות אחת אחרי השניה ברצף (הענק להם זמן התחלה וסיום) לפי המיון מקטן לגדול.

אלגוריתם חמדן: שיבוץ מטלות בת"ל

הרצאה: ישנן אוסף מטלות a1,a2,…,an ולכל מטלה זמן התחלה si וזמן סיום fi. המטרה: למצוא תת קבוצה בגודל מקסימאלי של מטלות לשיבוץ כך שלא יחתכו.

האלגוריתם סיבוכיות זמן : O(nlog(n)):

1. מיין את המשימות ע"פ זמני הסיום ממוקדם למאוחר.
2. $S\leftarrow\phi$. כל עוד יש משימות שעדיין לא טופלו:

-הוסף לפתרון S את המשימה a_i שמסתיימת ראשונה

-זרוק את כל המשימות שנחתכות עם a_i .