הטכניון - מכון טכנולוגי לישראל
הפקולטה למדעי המחשב

טלפון
איתי 29-4337
בית 25-5521
זיו 29-4260
בית 24-7054

מבנה נתונים 2

236312

מבחן סוף סמסטר

9 יולי, 1995
מרצה: א. איתי

משך המבחן: 72 שעות

את הבחינה יש לפתור **לבד** מבלי להשתמש בספרים או במאמרים פרט לאלה שאושרו במפורש.

נא לחתום על ההצהרה הבאה

אני מצהיר/ה בזה שלא נעזרתי באף אדם ובשום חומר מודפס, פרט ל-

Cormen Leiserson and Rivest

ולראיה באתי על החתום _____

יום _____ יולי1995

זכרו, הציון הוא יחסי, כך ששיתוף פעולה עלול לגרום להורדת ציונכם!

את התשובות יש לכתוב בצורה תמציתית, ממצה ומסודרת.
כל שאלה על עמוד נפרד, ולסדר את השאלות בסדר עולה.

בהצלחה

Fortune

Fifth Law of Applied Terror:
If you are given an open-book exam, you will forget your book.
Corollary:
If you are given a take-home exam, you will forget where you live.

שאלה 1

גאוני, עתודאי צעיר, מציע את האלגוריתם הבא למיון סדרה $x_1,\ldots,x_n$ (לא בהכרח שונים):
הכנס את x_1 ל-a[1].
עבור על שאר הסדרה איבר איבר והכנס אותם למערך a לפי הכלל הבא:
אם $x_i <$ a[1]
הזז את כל אברי a מקום אחד (למקומות a[2],...,a[i]) ושים את x_i במקום a[1].
אחרת
שים את x_i במקום a[i].

א. האם האלגוריתם של גאוני אמנם נכון? הוכח או תן דוגמא נגדית.
ב. כמה זמן לוקח האלגוריתם במקרה הגרוע?
ג. מהו הזמן הממוצע של האלגוריתם, בהנחה שכל הסדרים של הקלט הם שוי הסתברות?

שאלה 2

תהי H_1 קבוצת פונקציות ערבול אוניברסלית מ-A ל-B_1 ו-H_2 קבוצה כנ"ל מ-A ל-B_2.
נגדיר פונקציות $\langle h_i,h_j\rangle : A \to B_1 \times B_2$,

$$\langle h_i,h_j\rangle(x) = \langle h_i(x),h_j(x)\rangle$$

א. הראו שקבוצת הפונקציות $H = \{\langle h_i,h_j\rangle : h_i \in H_1, h_j \in H_2\}$ היא קבוצת פונקציות ערבול אוניברסלית מ-A ל-$B_1 \times B_2$.
ב. השתמשו ב-א' כדי להראות שלכל שתי קבוצות לא ריקות A ו-B קיימת קבוצת פונקציות ערבול אוניברסלית מ-A ל-B.

שאלה 3

יש לשכן עץ בינרי T בן n צמתים על דיסק. הנח שבסקטור ניתן לאחסן בדיוק 2 צמתים. כדי להגיע לצומת v יש לסרוק את כל המסלול מהשורש ל-v. זמן החיפוש פרופורציוני למספר הסקטורים <u>השונים</u> במסלול זה. שיכון I הוא מיפוי של הצמתים לסקטורים, יהי $\ell_I(v)$ מספר הסקטורים השונים במסלול זה. עליכם למצוא שיכון אופטמלי -- שיכון I, עבורו הערך הממוצע של $\ell_I(v)$ הוא מינימום. כלומר, שיכון הממזער את הפונקציה $\sum_{v\in T}\ell_I(v)$.
הניחו שעומד לרשותכם דיסק עם מספר בלתי מוגבל של סקטורים. אינכם נדרשים למזער את מספר הסקטורים הכולל בהם I משתמש.

<u>למה 1:</u> קיים שיכון אופטימלי I כך שאם שני צמתים u,v ממופים לאותו הסקטור, אזי u ו-v הם אב ובן.
<u>למה 2:</u> בשיכון אופטימלי סקטור שאינו מלא מכיל עלה.
<u>למה 3:</u> יהי I שיכון אופטימלי, ו-v צומת שאביו לא מופה לאותו סקטור ($I(v) \neq I(father(v))$).
אזי I הוא שיכון אופטימלי של התת-עץ ששורשו v.

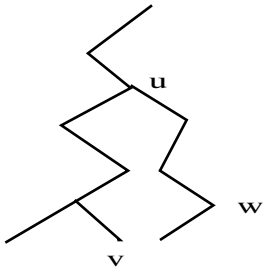
א. הוכיחו את הלמות.
ב. העזרו בלמות כדי לבנות אלגוריתם ליניארי למציאת שיכון אופטימלי.
<u>רמז</u>: השתמשו בתכנות דינמי.

שאלה 4

תהי S מחרוזת. עליך למצוא אלגוריתם ליניארי במקרה הגרוע למשימות הבאות:
א. למצוא את התת-מחרוזת בעלת אורך מכסימלי שמופיעה ב-S יותר מפעם אחת.
ב. בהנתן מספר i, מצא מחרוזת בעלת אורך $\leq i$ שמופיעה לפחות שלוש פעמים ב-S.
אפשר להסתמך על מבני נתונים שנסקרו בכיתה. אין צורך להוכיח תוצאות מהכיתה.

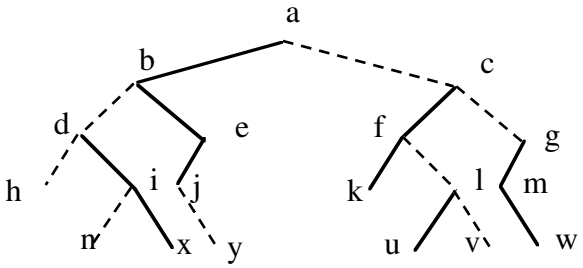
שאלה 5

יה י T עץ בינרי בן n צמתים. w,v צמתים של T. לצורך הדיון נכוון את הקשתות מהבנים לאבא והשורש יהיה למעלה.
צומת u הוא אב-קדמון נמוך ביותר של w,v ($u=lca(w,v)$) אם u הוא גם אב קדמון של v, וגם של w וכל צומת בעל תכונה זו הוא אב קדמון של u.



מטרת השאלה לפתח אלגוריתם למציאת אב קדמון נמוך ביותר.
האלגוריתם הפשוט ביותר הוא לסמן את צמתי המסלול מ-v לשורש ואח"כ לעלות מ-w לשורש עד אשר נתקלים בצומת מסומן. אם העץ שלם אלגוריתם זה לוקח $O(\log n)$. ברצוננו לפתח אלגוריתם למקרה שהעץ איננו שלם, למשל שרוך.

נחלק את קשתות T לשתי קבוצות: קשתות <u>שלמות</u> וקשתות <u>שבורות</u>. לכל צומת תכנס (מכיוון הבנים) לכל היותר קשת שלמה אחת, כך שהקשתות השלמות יהוו אוסף של מסלולים זרים בצמתים, שייקראו <u>מסלולים שלמים</u>. אם לא נכנסת לצומת ולא יוצאת ממנו אף קשת שלמה, הוא יהווה מסלול שלם באורך 0. כך שכל צומת שייך למסלול שלם יחיד.
<u>דוגמא</u>:



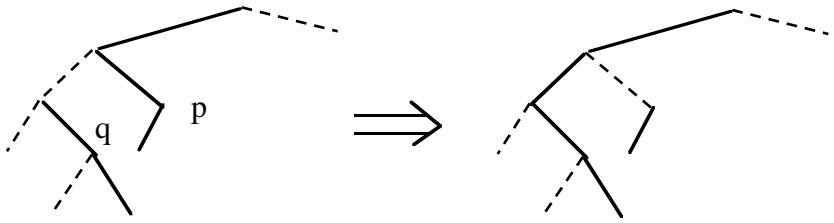
המסלולים השלמים הם: (h), (n), (x,i,d), (y), (j,e,b,a), (k,f,c), (u,l), (v), (w,m,g)

הצומת הגבוה ביותר של מסלול שלם ייקרא <u>ראש</u> המסלול (קרוב ביותר לשורש), והנמוך ביותר--<u>זנב</u> המסלול.
המסלולים שלמים ייוצגו ע"י מבנה נתונים המאפשר לבצע כל אחת מהפעולות הבאות בזמן $O(f(n))$:
path(v) מצא את המסלול השלם אליו שייך הצומת v.
head(p) מצא את ראש המסלול p.
concat(p,q) ניתן לביצוע על שני מסלולים שלמים p ו-q עבורם *father(head(q)) == tail(p)*.
הפעולה הופכת את הקשת מ-*head(q)* ל- *tail(p)* לקשת שלמה, כלומר מחליפה את המסלולים p,q במסלול חדש.

split(p,v) ניתן לביצוע אם *v* הוא צומת של *p*. הפעולה הופכת את הקשת השלמה שנכנסת ל-*v* לקשת שבורה, וע"י כך שוברת את *p* לשני מסלולים. (אם *v* היא הזנב של *p* , אז לפעולה זו אין שום אפקט).

א. הראה כיצד לממש את הפעולות בזמן $f(n) = O(\log n)$ במקרה בגרוע.

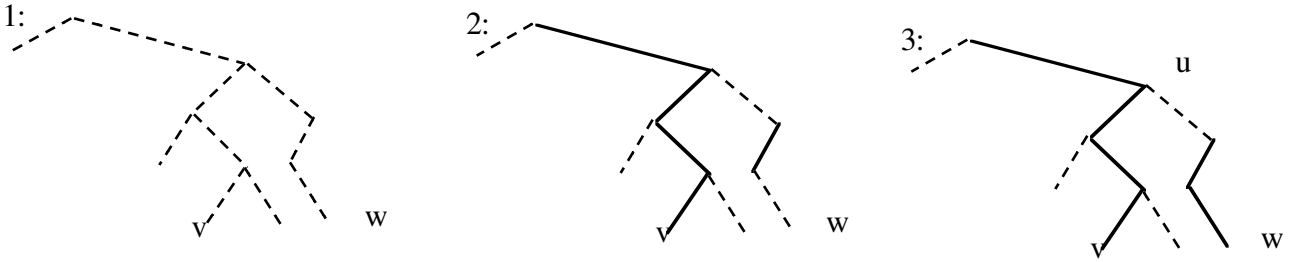
יהיו *p,q* מסלולים שלמים, כך ש- *v=father(head(q))* הוא צומת של *p*.
פעולת <u>הדבקה</u> (*glue*) תבצע *split(p,v)* ואח"כ *concat(p',q)* , כאשר *p'* הוא החלק העליון של *p* אחרי ה-*split*.



ביצוע *lca(v,w)*:
1. התחל מחלוקה חוקית כלשהי של הקשתות לשלמות ושבורות (למשל, כולן שבורות).
2. הפוך את כל המסלול מ-*v* לשורש למסלול שלם ע"י פעולות הדבקה.
3. בצע:

```
do forever{
   q = path(w);
   u = father(head(q));
   p = path(u);
   if(head(p) == root)
      return u;
   glue(p,q);
}
```

כלומר, נעבור על כל המסלול מ-w לשורש ונעצור כאשר נפגע במסלול (השלם) מ-v לשורש.



יהי *size(v)* מספר הצמתים בתת העץ ששורשו *v* (כולל *v*).
קשת *(v, father(v))* תקרא <u>כבדה</u> אם $size(v) > 2\,size(father(v))$, אחרת היא תקרא <u>קלה</u>.

ב. הוכח את הלמות והמשפט:
<u>למה 1</u>: לכל צומת v, המסלול מהצומת לשורש מכיל לכל היותר $\lfloor \log_2 n \rfloor$ קשתות קלות.
<u>למה 2</u>: לכל צמת נכנסת לכל היותר קשת כבדה אחת.
<u>משפט</u>: באלגי הנ"ל פעולת lca דורשת זמן משוערך $O(f(n)\log n)$.
ברור שזמן הביצוע פרופורציוני למספר ההדבקות, כך שדי להוכיח שבכל פעולת lca המספר (המשוערך) של הדבקות הוא $O(\log n)$.

<u>רמז</u>: $\varphi =$ – (מספר הקשתות שהן גם כבדות וגם שלמות)

שאלה 6

הראו סכימה חצי-משתמרת (partially persistent) לבניית פונקצית ערבול מושלמת חצי-דינמית (semi-dynamic) (אין הוצאות ואפשר להוסיף איבר רק לגירסה האחרונה).
כלומר, הפעולות הן:
insert(x) הוסף את x לגירסה האחרונה של הטבלה; זמן צפוי ל-t הכנסות O(t).
h(x,t) החזר את ערכה של פונקצית הערבול המושלם שהיתה קיימת אחרי ההכנסה ה- t.
זמן משוערך O(log n) עבור טבלה בת n איברים.
סיבוכיות מקום -- O(n).