

שאלה 1 (40 נק')

בשאלה שלפניכם מותר להשתמש ב radix sort וב bucket sort בלבד.

נתון מערך של m מספרים שלמים חיוביים $N_i > 0, 1 \leq i \leq m$. המספר N_i מיוצג ע"י d_i ספרות כאשר הספרה הכי סיגניפיקנטית (הראשונה משמאל) שונה מ-0, כך שארכם של מספרים שונים יכול להיות שונה זה מזה.

סה"כ הספרות המשמשות לייצוג המספרים הוא n , כלומר $\sum_{i=1}^m d_i = n$.

שני תלמידים נדרשו להציע אלגוריתם למיון המספרים. הראשון הציע להביא את ארכם של המספרים לאורך שווה (כמובן מבלי לשנות את ערכם) ואז למינם, ואילו השני הציע אלגוריתם הפועל ישירות על הייצוג הנתון של המספרים.

1. (5 נק') מהי סיבוכיות זמן הריצה בפתרון של התלמיד הראשון במקרה הגרוע? הסבירו הייטב כל צעד וצעד בפתרון ונמקו (הוכיחו) את תשובתכם.

מאחר ויתכן מצב שבו מספר הספרות באחד המספרים הוא $\Theta(n)$ ובשאר $\Theta(1)$, מספר הספרות הכולל שידרש לייצוג המספרים באורך שווה הוא $\Theta(mn)$, וזו גם סיבוכיות זמן המיון בשמוש ב radix sort.

2. (15 נק') התלמיד השני הציע פתרון המשתמש ב bucket sort תחילה ואחריו ב radix sort. הראו כיצד הפתרון המוצע ממין את מערך m המספרים בזמן $O(n)$. הסבירו הייטב כל צעד וצעד בפתרון ונמקו (הוכיחו) את תשובתכם.

מאחר ובכל מספר לפחות ספרה אחת, ולכל היותר $n - m$ ספרות בונים מערך של $n - m$ buckets ולתוכם זורקים מספרים שארכם שווה. ברור שכל המספרים ב bucket i קטנים מהמספרים ב bucket j , $1 \leq i < j \leq n - m$. המיון הזה לוקח $\Theta(m)$ זמן.

כעת ממינים בתוך כל bucket ע"י radix sort, מאחר וסך כל הספרות על פני כל ה buckets הוא n , השלב הזה דורש $\Theta(n)$ זמן. לבסוף משרשרים את הרשימות הממוינות בכל bucket מה bucket הקטן לגדול, שרשור שדורש $\Theta(n - m)$ זמן.

כעת מתבקשים למיין לקסיקוגרפית מערך של m מחרוזות (strings) $S_i > 0, 1 \leq i \leq m$. המחרוזת S_i מיוצגת ע"י d_i תווים, כאשר למחרוזות שונות יכולים להיות אורכים שונים (שימו לב שבמיון לקסיקוגרפי $a < ab < b < ba$).

סה"כ התווים המשמשים לייצוג המחרוזות הוא n , כלומר $\sum_{i=1}^m d_i = n$.

3. (20 נק') הציעו פתרון שממין את מערך m המחרוזות בזמן $O(n)$. אין צורך לרשום קוד אבל יש להסביר הייטב ובפירוט כל צעד וצעד בפתרון, לנמק ולהוכיח את תשובתכם.

הכוונה: ניתן לפתור את הבעיה ע"י שימוש רקורסיבי ב bucket sort.

הדבר אפשרי ע"י שימוש חוזר ונישנה (רקורסיבי) ב bucket sort.

1. כל bucket מוקדש לתו באלפאבית. בתום הסבב משמיטים את התו הראשון בכל מילה שב bucket. המילים הריקות (במידה וישנן) פירושן שבטרם ההשמטה של התו היו אלו המילים ה "קטנות" ביותר ב bucket (כולן בנות תו אחד ולכן זהות). מלים אלה מוכנסות לרשימה מקושרת בכל bucket.
 2. הסבב העוקב מתייחס לכל bucket מהסבב הקודם כבעייה עצמאית כנ"ל וממין כל תת-sub bucket. בכל כל sub-bucket משמיטים שוב את התו הראשון בכל מילה. אם נשארות מילים ריקות פירוש הדבר שבטרם ההשמטה של התו זו היו המילים ה "קטנות" ביותר ב שוב sub-bucket. משרשרים את כולן ע"פ סדר ה sub-buckets לרשימה שב parent bucket מסעיף 1.
 3. התהליך ממשיך באופן רקורסיבי עד שלא נשארות מילים.
 4. לבסוף משרשרים את את הרשימות שב 1 והתוצאה הינה רשימת המלים ממוינת.
- הסיבוכיות הינה $O(n)$ כי בכל איטרציה מושמטים תווים, וישנם לכל היותר n .

שאלה 2 (40 נק')

חברת "העץ הנדיב" ביקשה משלשה מהנדסים להציע אלגוריתם MST לחישוב עץ פורש מינימאלי T בגרף $G(V, E)$. בלי הגבלת הכלליות ניתן להניח שמשקלי כל הצלעות שונים זה מזה.

לפניכם שלשת האלגוריתמים המוצעים. לכל אחד מהם יש להוכיח באופן מלא ומדויק האם הוא אכן מוצא MST. אין צורך לדון בסיבוכיות האלגוריתמים.

(15 נק') אלגוריתם א'

1. ממיינים את הצלעות בסדר יורד של משקלן $w(e)$.
2. יהי $T = E$
3. לכל $e \in E$ הנלקחים בסדר היורד של משקלן
4. אם $T - \{e\}$ קשיר אזי
5. $T = T - \{e\}$
6. החזר את T

אלגוריתם א' מחזיר T שהוא אכן MST.

T המוחזר הינו קשיר. הוא גם חסר מעגלים משום שלו היה בו מעגל c אזי תהי $e' \in c$. הבדיקה בשורה 4 חייבת היתה לסלק את e' מ T , סתירה.

נניח כעת ש T איננו MST. ראשית, כל הקשתות של T שאינן על מעגל ב G חיוניות בכל עץ פורש, ובפרט MST. ולגבי השאר, נתבונן בקשת $T \cup \{e'\} \setminus e$ יוצר מעגל c .

ברור ש $w(e') > w(e), \forall e \in c - \{e'\}$. אחרת היתה איזושהי $e \in c - \{e'\}$ שהבדיקה בשורה 4 חייבת היתה מסלקת את e' מ T עוד בטרם נבדקת e . ולכן משקל כל הקשתות שאינן ב T ונמצאות על מעגלים ב G גדול ממשקלן של כל אלה ב T ושנמצאות על מעגלים ב G . ולכן T הינו MST.

(10 נק') אלגוריתם ב'

1. יהי $T = \emptyset$
2. לכל $e \in E$ הנלקחים בסדר שרירותי
3. אם $T \cup \{e\}$ ללא מעגלים אזי
4. $T = T \cup \{e\}$
5. החזר את T

אלגוריתם ב' איננו מחזיר T שהוא אכן MST.

יהי $G(V, E)$ גרף שלם. כל $e \in E$ נמצא על מעגל. מאחר וסדר הבחירה של $e \in E$ שרירותי, נבחר קשת ראשונה e זו שמשקלה מקסימלי. היא וודאי איננה חלק מאף MST, שכן משקל כל הקשתות האחרות על המעגלים ש e חלק מהם קטן יותר ולכן e ניתנת להחלפה ומשקל העץ קטן.

(15 נק') אלגוריתם ג'

1. יהי $T = \emptyset$
2. לכל $e \in E$ הנלקחים בסדר שרירותי
3. $T = T \cup \{e\}$
4. אם ל T ישנו מעגל c אזי
5. יהי $e^* = \operatorname{argmax}_{e \in c} w(e)$
6. $T = T - \{e^*\}$
7. החזר את T

אלגוריתם ג' מחזיר T שהוא אכן MST.

T המוחזר הינו חסר מעגלים משום שלו היה בו מעגל c אזי תהיה $e' \in c$. הבדיקה בשורה 4 חייבת היתה לסלק את e' מ T , סתירה. T גם מקסימלי, כי אם לא אז ניתן להוסיף לו קשת שכבר סולקה קודם בבדיקה בשורה 4 כאשר T היה חלקי. מסקנה, T אכן עץ פורש.

T גם MST. ראשית, כל הקשתות של T שאינן על מעגל ב G חיוניות בכל עץ פורש, ובפרט MST. ולגבי השאר, נתבונן בקשת $T \cup \{e'\} \setminus e$ יוצר מעגל c .

ברור ש $w(e') > w(e), \forall e \in c - \{e'\}$, כי e' סולקה בבדיקה בשורה 4 משום שהיתה הכבדה ביותר במעגל שיצרה. ולכן משקל כל הקשתות שאינן ב T ונמצאות על מעגלים ב G גדול ממשקלן של כל אלה ב T ושנמצאות על מעגלים ב G . ולכן T הינו MST.

שאלה 3 (40 נק')

נתונות שתי שפות $A: \langle a_1, a_2, \dots, a_n \rangle$ ו $B: \langle b_1, b_2, \dots, b_n \rangle$ המכילות מספר זהה של מלים כאשר לכל מילה בשפה A ישנו תרגום אחד ויחיד למילה בשפה B . המלים ב A ממוינות בסדר מילוני כך ש $a_1 < a_2 < \dots < a_n$

חברת "תרגומון" נדרשה לתכנן אלגוריתם לתרגום מ A ל B והטילה את המשימה על שלשה מהנדסים שנדרשו להציע מבנה נתונים המתאים למילון. לכל אבר במילון ישנו מפתח שהוא המילה בשפה A ובצמוד אליה התרגם לשפה B .

מהנדס א' הציע להשתמש ברשימה מקושרת, מהנדס ב' הציע להשתמש בעץ חיפוש בינארי ומהנדס ג' בעץ AVL. נתון טקסט בשפה A שגודלו m מילים.

1. (5 נק') הסבירו מהו הזמן $t(m, n)$ הנדרש לתרגום במקרה הטוב ביותר, הממוצע והגרוע ביותר בכל אחד ממבני הנתונים המוצעים. נדרש להסביר כל חישוב.

$t(m, n)$		רשימה מקושרת	עץ חיפוש בינארי	עץ AVL
טוב ביותר	חישוב	$\Theta(m)$	$\Theta(m)$	$\Theta(m)$
	הסבר	בקובץ רק המילה הראשונה ברשימה	בקובץ מופיעה רק המילה שבשורש	בקובץ מופיעה רק המילה שבשורש
ממוצע	חישוב	$O(mn)$	$O(m \log n)$	$O(m \log n)$
	הסבר	אורך חיפוש מילה $O(n)$	עומק העץ $O(\log n)$	עומק העץ $O(\log n)$
גרוע ביותר	חישוב	$O(mn)$	$O(mn)$	$O(m \log n)$
	הסבר	בקובץ רק המילה האחרונה ברשימה	עומק העץ $O(n)$ ובקובץ רק המילה העמוק ביותר	עומק העץ $O(\log n)$

מניחים שהשכיחות $p_i = \Pr(a_i)$ בשפה A ידועות מראש וכן $\sum_{i=1}^n p_i = 1$.

2. (5 נק') האם ניתן להשתמש בשכיחות המלים בכל אחד ממבני הנתונים שב 1 לקטנת $t(m, n)$? ובמידה וניתן אז הסבירו כיצד.

כן. בכל אחד ממבני הנתונים המצאות מלים בעלות שכיחות גבוהה קרוב לראש הרשימה או קרוב לשורש תקטין את מספר הצעדים למציאת המילה.

השאלות הבאות מתייחסות לזמן התרגום עבור טקסט נתון שבו שכיחויות המילים ידועות מראש. כל הזמנים שלהלן מתייחסים לזמן חיפוש ממוצע למילה.

יהי T עץ חיפוש בינארי שעבורו זמן החיפוש הינו מינימלי. מטרת השאלות הבאות היא לגלות את המבנה של T ע"י פיצול העץ באופן רקורסיבי לתתי עצים ימני ושמאלי באופן אופטימאלי.

הכוונה: הבעיה ניתנת לפתרון בתכנות דינאמי. ניתן להעזר ברעיונות מקוד HUFFMAN ומציאת קינון סוגרים אופטימלי בשרשרת כפל מטריצות (שני הנושאים נלמדו בקורס).

3. (2 נק') יהי $T_{i,j}$ תת עץ של T עבור המפתחות $a_i < a_{i+1} < \dots < a_j$, $1 \leq i \leq j \leq n$. האם זמן החיפוש ב $T_{i,j}$ מינימלי גם כן? הסבירו את תשובתכם.

כן. אחרת ניתן היה להחליף את $T_{i,j}$ בתת עץ אחר $T'_{i,j}$ טוב יותר בסתירה לאופטימליות של T .

4. (3 נק') יהא המפתח a_r מצוי בשורש של $T_{i,j}$, $i \leq r \leq j$. מה ניתן לומר על זמני החיפוש ב $T_{i,r-1}$ ו $T_{r+1,j}$? הסבירו את תשובתכם.

זמני החיפוש בהם חייבים להיות מינימלים גם כן. אחרת ניתן היה להחליף את $T_{i,r-1}$ בתת עץ אחר $T'_{i,r-1}$ טוב יותר בסתירה לאופטימליות של $T_{i,j}$. כנ"ל ל $T_{r+1,j}$.

5. (5 נק') רישמו ביטוי לשכיחות $q(a_r)$ של ביקור בצומת שבו נמצא המפתח a_r והסבירו את הביטוי.

מאחר ומציאת מילה בכל תת העץ שבשורש שלו נמצא המפתח a_r (כולל הקדקד) מחייבת מעבר דרך השורש, $q(a_r) = \sum_{k=i}^j p_k$.

נסמן $p_{i,j} = \sum_{k=i}^j p_k$. נסמן ב $t[i,j]$ את זמן חיפוש הממוצע המינימאלי למילה עבור המפתחות $a_i < a_{i+1} < \dots < a_j$, $1 \leq i \leq j \leq n$.

6. (10 נק') רישמו את הביטוי לחישוב $t[i,j]$ (הסתמכו על תשובותיכם ב 3-5).

$$t[i,j] = \begin{cases} p_i & i = j \\ \min_{i \leq r \leq j} \{t[i, r-1] + p_{i,j} + t[r+1, j]\} & i < j \end{cases}$$

7. (10 נק') בהסתמך על 4-7 הציעו אלגוריתם מיטבי, שזמן חישובו חסום פולינומיאלי, לחישוב $t[1, n]$. אין צורך לרשום קוד וניתן להסתפק בהסבר מילולי. מהי סיבוכיות זמן הריצה של האלגוריתם שהצעתם?

הביטוי ב 7 ניתן לפתרון רקורסיבי אולם ע"פ 3-5 הבעיה מקיימת את כל הדרישות לפתרון בתכנות דינאמי: אופטימליות של תתי פתרונות, חזרה על אותן תתי בעיות ושימוש ברישום של פתרונות קודמים (memoization). זוהי בעיה דומה מאד למציאת סדר החישוב האופטימלי

בשרשרת של כפל מטריצות. ע"י פתרון bottom-up המשתמש בשלש לולאות מקוננות ניתן לבנות את עץ החיפוש האופטימאלי בזמן $O(n^3)$.