1. א. תשובה: O(m),

ראשית ישנן m+1 הכנסות כאשר כל אחת הן O(1) O(m).

לאחר מכן מתבצע deleteMin(), מוחקים את -1 זו פעולה ב O(1) ואז עושים קונסולידציה על m עצים בגודל 1. מכיוון ואנחנו תמיד נכניס עץ בגודל 1, בכל הכנסה שניה יתבצע איחוד בין שני עצים בגודל 1, באותו אופן בכל הכנסה רביעית יתבצע איחוד בין 2 עצים בגודל 2 (ובפרט בגודל 1 אך ספרנו זאת כבר). באופן כללי יתבצע לעץ בגודל איחוד בכל הכנסות כאשר הגדול ביותר יהיה log(m)-1 פעם אחת בהכנסה האחרונה. בנוסף כל איחוד הינו בO(1) ללא תלות בגודל של העצים שאנו מאחדים, כי אנחנו רק מחברים שורשים. לכן לסיכום עלות הפעולה תהיה:

עכשיו נבצע log(m) פעולות של decrease key לאיברים שהינם אי זוגיים, מהטענה מקודם בכל הכנסה i אי זוגית כזו כבר יהיה איבר בתא של העצים בגודל 0 והערך בו יהיה עם מפתח i-1 ולכן הוא יהיה הבן של i-1 כי זה האחרון שהכנסנו ובפרט עלה, וכך ישאר מכיוון ואם מתבצע לינק זה אך ורק בין שורשים וi אינו כזה ולא יהיה כזה במהלך האיחודים.

עכשיו כשאנחנו מבצעים decrease key של m+1 לצומת הוא אי זוגי ולכן עלה בהכרח יהיה קטן יותר מאביו ולכן יתבצע ניתוק, נסמן את אביו ונשים לב שלכל צומת אי זוגי אביו זה הקודם (קודם כהופכי של עוקב) שלו ובפרט לכל צומת אי זוגי אביו שונה, ולכן יתבצע ניתוק כשהאבא לא מסומן ולכן יהיה ניתוק יחיד והפעולה תעלה O(1). ולכן סדרת הניתוקים תעלה O(log(m)).

לסיכום:

ב.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Potential** | **Total cuts** | **Total links** | **Runtime (ms)** | **m** |
| 29 | 10 | 1023 | 6 |  |
| 44 | 15 | 32,767 | 34 |  |
| 59 | 20 | 1,048,575 | 265 |  |
| 74 | 25 | 33,554,431 | 6,487 |  |

נצרף ראשית טבלה המסכמת את סעיפים ג-ו

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decrease key max cost | Potential | Total Cuts | Total Links | Case |
| (Skip) |  | Log(m) | m-1 | (c) original |
| (Skip) | 1 | 0 | m-1 | (d) deckey( |
| (Skip) | m+1 | 0 | 0 | (e) remove line #2 |
| Log(m)-1 | 2log(m) | 2Log(m)-1 | m-1 | (f) added line #4 |

ג. כמות הcuts הינה log(m), ישנם log(m) איברים שאותם ננתק מהוריהם, וכפי שהסברנו בסעיף א' כל אחד מהם גורם לחיתוך יחיד מכיוון ולכל אחד הורה שונה.

הפוטנציאל הינו , הפוטנציאל בהגדרה הינו נסביר כל גורם בנפרד:

Marked = log(m)-1: אנו מנתקים log(m) איברים כאשר לכל אחד אבא שונה, ומכיוון ויש בסוף איברים לאחר פעולת הconsolidate נקבל עץ בינומי מלא. כאשר שורשו הינו 0, ובנו הוא 1 (1 הוכנס ישר אחרי 0), לכן כשנמחק את 1 לא נסמן את 0, כי הוא שורש. ובאחרים האבא לא שורש וכן נסמן אותו. לכן יהיו log(m)-1 מסומנים.

Trees=log(m)+1: מכיוון וכל decrease key גורם לחיתוך אנחנו יוצרים log(m) עצים נוספים וישנו את העץ שאיתו התחלנו ולכן נקבל log(m)+1 עצים.

נציב בנוסחה ונקבל כנדרש.

מספר החיבורים יהיה כמספר הקשתות בעץ מכיוון ואנחנו מוסיפים לינק כאשר אנחנו יוצרים קשר בין אבא ובן (ואנחנו עושים זאת פעם אחת). אנחנו מקבלים בסוף עץ בינומי מלא, ונוכיח באינדוקציה כי לעץ כזה יש קשתות. K=0 ישנו צומת אחד ולכן אין קשתות,

נניח נכונות על k ונוכיח עבור k+1, ראינו כי עץ בינומי אלו שני עצים מדרגה k כאשר שורש אחד בן של השורש השני, בשני העצים ביחד יש ובנוסף יש את הקשת המחברת בין השורשים ונקבל כנדרש. ולכן בערימה שלנו נקבל m-1 חיבורים.

ד. נשים לב כי אנחנו מורידים מהאיברים הבאים: נסמן k=log(m)

ניתן הסבר למה כל מפתחות אלה הם בנים אחד של השני (משמאל לימין) בעץ הסופי:

נשים לב שלכל מספר כזה הוא המספר הגדול ביותר בין 0 ל m-1 (הערכים בערימה) אשר מתחלק ב , אחריו יבוצעו רק אחודים עד דרגה מכיוון וזה האיבר ה שנכנס (כי אנחנו מכניסים את 0), וכמו שהסברנו בסעיף א' מתבצעים אחודים עד דרגה j רק בהכנסות המתחלקות ב () ולכן גם ההכנסה לפניו ביצעה איחוד עד דרגה i. ובפרט הוא יכנס כשורש יחיד, מכיוון וכל איבר שיכנס אחריו גדול ממנו הוא יהיה שורש ומכיוון וישנן עוד הכנסות שהן  *(כי ישנם מספרים יותר גדולים שיתחלקו בזה) ולכן לפני ההכנסה האחרונה הוא יהיה שורש מדרגה i-1 (כי ביצענו איחוד עד דרגה i-1). וזה נכון לכל i ולכן לפני ההכנסה האחרונה שתבצע איחוד בין כל הדרגות (כי היא ההכנסה במספר ה- ), הם יהיו השורשים ולכן בנים אחד של השני לפי הסדר.*

*בהורדת המפתח אלה אנחנו מורדים לכולם את אותו ערך, אנחנו מתחילים בערך הכי קטן, ולאחר מכן בבנו שקטן ממנו וכן הלאה (מההסבר מקודם), ולכן אנחנו תמיד נוריד לערך שגדול מהאבא כי הקטנו אותו קודם לכן. ולכן לא יהיו חיתוכים כלל ובפרט לא סימונים.*

*לכן total cuts=0.*

*פוטנציאל =2\*0+1 כי ישנו עץ אחד.*

*ומספר החיבורים לא ישתנה מסעיף קודם כי לא איחדנו שום דבר נוסף.*

*ה. מכיוון ולא מבצעים מחיקה למינימום לא יתבצע קונסולידציה ולכן כלל הצמתים ישארו כעץ מדרגה 0, כולל -1 שלא מחקנו. ישנם m+1 עצים. ובכל הורדת מפתח לא יתבצע כלום (למעט הורדת המפתח עצמו) כי הם שורשים ולא ניתן לנתק או לסמן.*

*מכיוון ולא התבצעה קונסולידציה אין לינקים שתבצעו ולכן 0.*

*מכיוון ולא חתכנו כלום אין חיתוכים.*

*מכיוון ויש m+1 עצים ואין סימונים הפוטנציאל הינו m+1.*

*ו. total links- מכיוון והפעולה האחרונה לא עושה לינקים לא ישתנה מספר הלינקים מהמקרה המקורי.*

*נשים לב כי m-2 הינו האיבר האחרון בשרשרת שהוסברה בסעיף ד', וכמו שראינו כל איבר בשרשרת נכנס כשורש בדרגה 0 ולכן האיבר שנכנס אחריו (הם האיברים שנמחקים בשורה 3, כי הם +1 האיברים בשרשרת מסעיף ד'), ולכן כאשר נבצע את הורדת המפתח המתבקשת נוריד את ערך המפתח ל -3 (שכמובן זה לא תקין) ולכן יתבצע חיתוך, ומכיוון וכל האיברים בשרשרת הם מסומנים אחרי פעולות הורדת המפתח הקודמות, לכן יבוצע cascading cuts עד השורש (לא כולל) ולכן יבצעו log(m)-1 חיתוכים כי אורך השרשרת הוא log(m).*

*ולכן סך כל החיתוכים עד כה יהיה:*

*וברור כי מספר החיתוכים הגדול ביותר הוא log(m)-1 נובע מההסבר לעיל.*

*מספר העצים יגדל ב log(m)-1 כי אנו מבצעים log(m)-1 חיתוכים, וכל אלה גם כן מסומנים ולכן מספר המסומנים קטן ב log(m)-1 ולכן סך הכל הפוטנציאל ירד ב log(m)-1 (כי בפוטנציאל המסומנים שוקלים פי 2 מהעצים).*

*ונקבל 2log(m).*

1. *א.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Potential* | *Total cuts* | *Total Links* | *Runtime(ms)* | *m* |
| *6* | *0* | *723* | *14* | *728* |
| *6* | *0* | *6555* | *28* | *6560* |
| *9* | *0* | *59040* | *289* | *59048* |
| *10* | *0* | *531431* | *782* | *531440* |
| *14* | *0* | *4782955* | *4360* | *4782968* |

*ב. ראשית הכנסת המפתוח תעלה O(m). ולאחר מכן המחיקה הראשונה תעלה O(m) כי יש לנו m עצים שונים. לאחר מכן, ראינו בהרצאה שכל פעולה כזו תעלה באמורטייז O(logm) מכיוון שהערימה אחרי קונסולידציה. ישנם O(m) פעולות מחיקה, ולכן סך הפעולה תעלה O(mlogm+m)=O(mlogm)*

*ג. ישנן 0 פעולות cut מכיוון ופעולה זו מבוצעת אך ורק בdecrease key ואנו לא מבצעים פעולה זו.*

*מספר הלינקים: m-(#number of 1's in the binary representation of m) (m פחות משהו)*

*במקום לכתוב את זה נכתוב במהלך הפתרון #bits*

*ראשית מכיוון ואנו מכניסים m+1 איברים, לאחר המחיקה הראשונה תתבצע קונסולידציה לכלל האיברים כאשר נתחיל מהגדולים ביותר ונסיים ב 1 כי הכנסנו לפי הסדר והאחרון שהוכנס הוא הראשון.*

*ונקבל עצים בינומים, כי אם בערימה רק עצים בינומים (אמתיים) זה יתנהג כמו ערימה עצלה ומכיוון ואין decrease key וחיתוכים בפרט אז העצים יישארו בינומים לאורך כל חיי התרגיל.*

*בקונסולידציה הראשונה אנחנו יוצרים מספר עצים כמספר האחדות של m בייצוג בינארי (כי עתה יש m איברים וזו ערימה בינומית מלאה), וראינו בהרצאה כי מספר העצים הוא מספר האחדות בייצוג הבינארי. ונשים לב כי לבניית כל עץ בוצעו לינקים כגודל העץ פחות 1, נוכיח באינדוקציה:*

*גודל העץ =1, 0 לינקים.*

*נניח נכונות לעץ בגודל K/2 (נשים לב שעצים בינומים הם חזקות של 2 [הוכח בשיעורי בית]) ונוכיח עבור k, עץ בגודל k הינו חיבור של שני עצים בגודל k/2; לכל אחד היה צריך k/2-1 לינקים לפי הנחת האינדוקציה לכן היו עד כה k-2 לינקים ועוד לינק נוסף כדי לחבר בין השורשים ונקבל k-1 לינקים.*

*לכן כמות הלינקים של סך העצים בערימה לאחר המחיקה הראשונה תהיה m-#trees מכיוון שכל עץ מוריד לינק 1 וראינו כי #trees=#bits.*

*עכשיו נוכיח כי בכל מחיקה נוספת המינימום יהיה שורש העץ הכי קטן, נשים לב כי במהלך הקונסולידציה הראשונה בכל שלב, האיבר המינימלי הוא שורש העץ הקטן ביותר, כי הוא האחרון שנכנס ואם נבצע איחודים לא יהיו עצים קטנים משלו (כי הוא תמיד הקטן).  
עתה, בכל פעם שאנחנו נבצע delete min כאשר המינימום הוא שורש העץ הקטן ביותר זה כאילו חזרנו צעד בקונסולידציה. כאשר הכנסנו איבר זה בקונסולידציה הוא איחד מספר עצים בדרגות שונות שכולם קטנים משאר העצים בעץ כי העץ הנוכחי הוא בעל הדרגה הקטנה ביותר ודרגות הבנים שלו קטנות יותר. ולכן לא יתבצעו אף לינקים במחיקה זו, וכעת הערימה תיראה שלב אחד לפני כן בתהליך הקונסולידציה (לפני שאיחדנו אותם בתהליך עם המינימום).*

*בערימה שלנו מכיוון ואנחנו מכניסים לפי הסדר, בקונסולידציה זה יהיה בסדר ההפוך, לכן בכל שלב האחרון שהכנסנו בקונסולידציה (המינימום באותו שלב) יהיה שורש העץ הקטן ביותר. לכן, מההסבר לעיל, בתהליך המחיקות שלנו בכל שלב המינימום יהיה שורש העץ הקטן ביותר ובנוסף מאותו הסבר לא יתבצעו עוד לינקים. לכן בתהליך יבוצעו רק הלינקים מהמחיקה הראשונה שעלו כאמור m-#bits.*

*נשים לב מכיוון ו i זוגי אז*

ובפרט m מתחלק ב 8 כלומר הייצוג הבינארי שלו מסתיים ב 000. מכיוון לאחר כל המחיקות נישאר עם , משום שאנחנו מתחילים עם m+1 איברים ומוחקים , בגלל שחלוקה ב 4 של מספר המתחלק ב4 זה כמו לעשות הזזה ב2 ימינה bitwise, ולכן ה יסתיים ב0 וגם יהיו לו את אותו מספר 1 בייצוג הבינארי כמו m, ומכיוון ואנו מוסיפים 1, אז נוסף עוד 1 למספר האחדים (במקום ה0 בסוף). בגלל שאנחנו מקבלים ערימה בינומית מלאה בסוף התהליך מספר העצים בה הוא מספר האחדים בייצוג הבינארי של שהוא #bits+1 ומכיוון שאין חיתוכים אז אין צמתים מסומנים ולכן הפוטנציאל הוא מספר העצים כלומר:

Potential= #bits +1