

מבנה Union Find

- מבנה נתונים המאפשר את הפעולות הבאות
 - הכנסה (Insert) – צור סט ובו האבר החדש
 - איחוד (Union) – אחד שני סטים קיימים
 - חיפוש (Find) – בהינתן אבר מצא את הסט אליו הוא שייך והחזר אבר המייצג את הסט (Representative)



http://en.wikipedia.org/wiki/Jimmy_Hoffa

מבני נתונים 08a

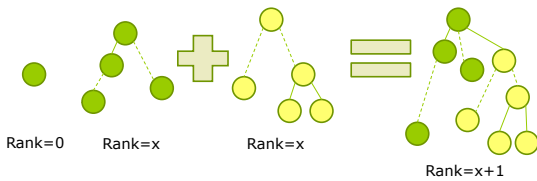
תרגול 14
20.3.2008

Union-Find & Suffix Trees

ליאור שפירא

מימוש Union Find

- מה יעילות המבנה במימוש ע"י עצים?
 - מימוש נאיבי – כמו רשימות מקושרות. למה?
 - שיפורים שניתן לעשות
 - Union By Rank



מימוש Union Find

- מימוש ע"י רשימות מקושרות
 - כל סט יהיה רשימה מקושרת
 - איחוד שני סטים הינו בעצם שרשרת הרשימות
 - חיפוש לוקח זמן לינארי
- מימוש ע"י עצים
 - כל סט נמצא בעץ (מבנה הנתונים כולו – יער)
 - נציג כל סט יהיה השורש של העץ
 - איחוד ע"י חיבור שני עצים (הרכבת אחד על השני)
 - חיפוש ע"י מציאת האבר וטיפוס בעץ עד לשורש



מימוש Union Find

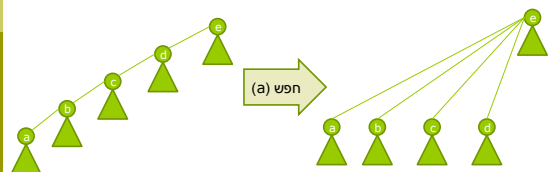
- מה יעילות המבנה במימוש ע"י עצים?
 - מימוש נאיבי – כמו רשימות מקושרות. למה?
 - שיפורים שניתן לעשות
 - Union By Rank – משפר את היעילות של חיפוש לזמן $O(\log n)$
 - כיווץ מסלולים (Path Compression) – משפר את היעילות של הפעולות כך: עבור n פעולות הכנסה (יצירת סט) f - פעולות חיפוש נקבל יעילות $O(n + f \cdot (1 + \log_{2,f} n))$
 - מה אם נפעיל את שתי השיטות יחדיו?
 - החסם על סדרת זמן פעולות הינו $O(m \cdot \alpha(n))$

$$O(m \cdot \alpha(n))$$

הפ' ההופכית לפ' אקרמן, זו פ' שגדלה מאד מאד לאט... עבור רוב השימושים של union find ערכה קטן מ-4

מימוש Union Find

- מה יעילות המבנה במימוש ע"י עצים?
 - מימוש נאיבי – כמו רשימות מקושרות. למה?
 - שיפורים שניתן לעשות
 - Union By Rank – משפר את היעילות של חיפוש לזמן $O(\log n)$
 - כיווץ מסלולים (Path Compression)



תרגיל 1

בהינתן מבנה נתונים של union find (הכולל את השיפורים של m פעולות (union by rank & path compression), ניתנת סדרה של m פעולות (make-set, union, find) ובה כל פעולות ה-union נעשות לפני פעולות ה-find. הראו שזמן הביצוע סדרה זו הוא לינארי $O(m)$

- פתרון
- מהו החסם העליון? $O(m \cdot \alpha(n))$
 - נפתור בשיטת הבנק



תרגיל 1

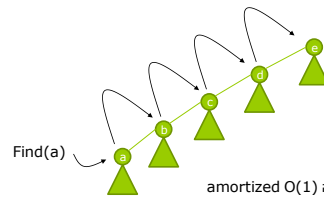
פתרון בשיטת הבנק

- כדי לסדרה של m פעולות תיקח זמן $O(m)$ כל פעולה צריכה לעלות מחיר קבוע
- הפעולות הזולות ישלמו על היקרות
- אז מי הפעולות היקרות?
- Make-set
- Union
- Find

תרגיל 1

פתרון בשיטת הבנק (המשך)

- נרצה לשלם על פעולות find בעזרת הפעולות הזולות
- בכל פעולת union נשים מטבע על השורש של הסט החדש
- בפעולת find נטפס ונעשה path compression
- כל פעולה תכווץ שורש שנעשה עליו union ולכן יש שם מטבע



תרגיל 2 (+-)

בהינתן גרף $G(V,E)$, מצאו את רכיבי הקשירות של הגרף

פתרון?

- נקרא ל-make set על כל קדקוד של הגרף
- נעבור על כל הקשתות ולכל אחת נקרא ל- $Union(find(u), find(v))$ כאשר u ו-v צמתי הגרף
- כל סט במבנה הנתונים מכיל כעת רכיב קשירות יחיד
- סיבוכיות $O((|V|+|E|) \cdot \alpha(|V|))$

תרגיל 3

נגדיר משחק HEX על לוח משושים, כל שחקן צובע בתורו משבצת בלבן/שחור (תלוי בשחקן). מנצח השחקן שמצליח לחבר ראשון את שני הצדדים שלו

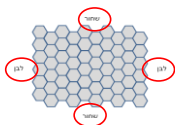


כיצד נחזיק ביעילות את מבנה הלוח כך שזמן העיבוד כל תור ייקח $O(\log n)$ ונוכל לומר אם אחד השחקנים ניצח?

תרגיל 3 - פתרון

נגדיר מבנה union find ובו ארבע אברים התחלתיים

- White-right
- White-left
- Black-top
- Black-bottom



בכל תור נעשה

- $Insert(X)$ כך ש-X משבצת לבנה או שחורה
- Union של X עם כל שכן שנוגע בו באותו צבע
- מה תנאי הסיום?
- $Find(W-R) = Find(W-L)$ אזי הלבן ניצח
- $Find(B-T) = Find(B-B)$ אזי השחור ניצח

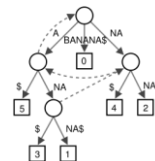


תרגיל 1

בנו suffix tree למחרוזת הבאה

Supercalifragilisticexpialidocious

Suffix Trees



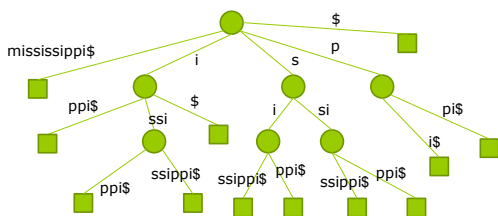
תרגיל 1

בואו נבנה את העץ קודם

$i, pi, ppi, ippi, sippi, ssippi, issippi, sissippi, ssissippi, ississippi, mississippi$

And sorted:

$i, ippi, issippi, ississippi, mississippi, pi, ppi, sippi, sissippi, ssippi, ssissippi$



תרגיל 1 באמת

תארו אלגוריתם הרץ בזמן ליניארי המקבל כקלט מחרוזת S ומוצא את תת המחרוזות הקצרה ביותר החוזרת פעם אחת בלבד ב-S.

למשל mississippi

תרגיל 2

נגדיר תת מחרוזת z באורך k להיות חזרה מכסימלית ימנית במחרוזת S אם קיימים i ו-j, $i \neq j$ כך ש:

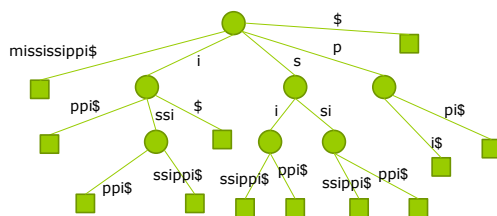
$$z = s_i s_{i+1} \dots s_{i+k-1}$$

$$z = s_j s_{j+1} \dots s_{j+k-1}$$

$$s_{i+k} \neq s_{j+k}$$

תאר אלגוריתם ליניארי המקבל כקלט מחרוזת S ומספר k ומחשב את כל החזרות המכסימליות הימניות באורך k. עבור כל חזרה האלגוריתם מדווח את האינדקס בו מתחיל המופע הראשון שלה.

תרגיל 1

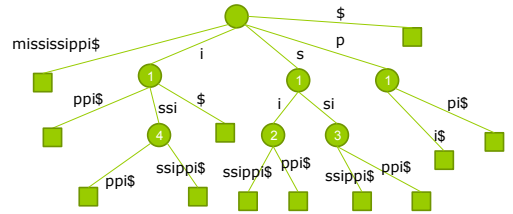


מחרוזת החוזרת פעם אחת בלבד חייבת להסתיים בעלה נחפש את הצומת k כך שיש לה בן עלה (שאינו \$) ועומקה מינימלי. התשובה הינה המחרוזת מהשושור עד לצומת k ועוד התו הראשון על הקשת מהצומת לעלה

התשובה היא m ←

תרגיל 2

□ בואו נסתכל שוב על העץ של mississippi:



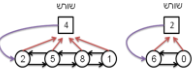
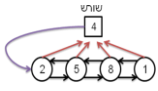
מחרוזת באורך החזרת מס' פעמים במקומות שונים, איך תופיע בעץ?

צומת פנימית בעלת מס' בנים, בעומק k ←

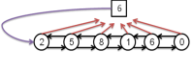
אם יש זמן...

□ להלן מבנה נתונים ל-union find:

- כל קבוצה תחזק ברשימה משורשרת דו-כיוונית, כל אלמנט מצביע לצומת מיוחד (שורש) המייצג את הקבוצה



↓
דומה ל-union:



- Find מקבלת מצביע ומחזירה מצביע לשורש
- Union מוסיפה קב' קטנה לגדולה ומעדכנת את השורש

אם יש זמן

- כמה זמן תיקח סדרת מ פעולות כאשר נתחיל מאוסף של n קבוצות בעלות אבר אחד (כל אחת). הנח $n > m$

הסוף...