

Records Company

Records Company

hashTable m-customers

UnionFind m-records

AVLTreeExtra<int, Shared_ptr<customer>> m_clubMembers

hashTable

int m-numberOfObjects

int m-tableSize

HashFunc m-hashFunc

AVLTree<int, Shared_ptr<customer>>* m-table

UnionFind

int m-numberOfRecords

CDRecord* m-records

nodeInGroup** m-recordsMappedToGroup

GroupList m-groups

nodeInGroup

nodeInGroup* m-father

Group* m-group

CDRecord m-record

int m-extraForHeightCalc

int height

GroupList

Group* root

Group

int m-column

int m-height

nodeInGroup* m-base

int m-size

Group* m-prev

Group* m-next

(new) m-purchases, (copy) m-copies, (new) m-id : shown for Group object : CDRecord (*)

(new) m-is-member, (copy) m-phone, (new) m-id : shown for Customer object : customer (*)

(new) m-monthly-expenses, (copy) true, (new) m-

יְהוָה יְהוָה אֱלֹהִים יְהוָה

המבנה היררכי של נתונים ועיבודם : AVL פג (1)

לעתה נראה כיצד תבצע חישוב ייחודי ל- h המבוסס על הערך x ו- m . מושג זה נקרא $h(x)$.

לעתות ה- α מוגדר כטבילה ב- $\frac{1}{\alpha}$ (טבילה יתירה (טבילה נוספת תוא), טבילה נוספת תוא) ו- $\alpha = 2$

. $O(1)$ - נס מוכן . באתן hash יתבצע : init() ברגע

AVL tree insert operation. Inserting data into a binary search tree involves finding the correct position for the new key and then inserting it while maintaining the tree's balance.

ו' אינטגרל $\int \frac{1}{\sqrt{n}} dx$ נבדקה הפעם (לעתות מילוי ועקבות)

(1) የዕለታዊ ስራውን በዚህ የዕለታዊ ስራውን አካል ተስፋይ ይችላል

• Key new for each update : find(key)

אנו יכולים לומר ש- $O(1)$ מוגדרת כפונקציית זרימה.

ב- $O(\log n)$ -) (תואם לתאום ה- $\Theta(n \log n)$ ב- $O(n \log n)$ -)

מבחן שאלות - בואנו לחקור מושג key, וה, ה, extra nodes מהן AVL-tree יתאפשר:

כדי לשבור תר על כל מילויו שלילית ותאך דינג (נפטר לפרטן הטעסלאה נטינג ו-טנט).
(13)

למי של מילוי ע"י מילויים נוספים, מילויים נוספים יתאפשרו על מנת לשלב מילויים נוספים.

טער קריין (הנערין) בז' טדרן (טדרן)

ይህንን በinit() የሚያስፈልግ ይችላል.

$O(1)$ je něco

• សម្រាប់ការស្វែងរកនៅលើបញ្ជីដែលត្រូវបានស្វែងរក : `find(key)`

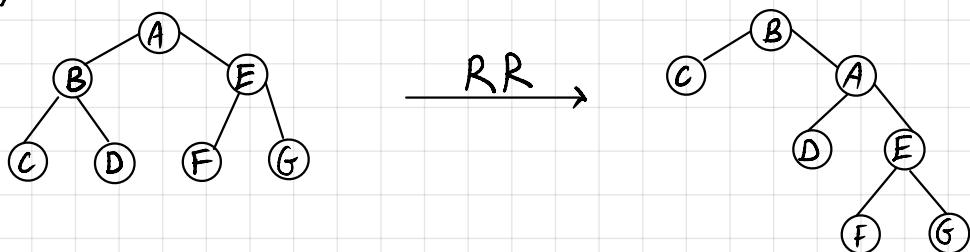
$O(\log n)$: It's about

Key search gets O(log n) time complexity, but sum of extra values -> $O(n)$ time complexity: sum_extras(key)

מזהה מילויים בפונקציית `add_extra(value, key)` בפונקציית `logon` שבדף הקודם. פונקציית `logon` מקבלת שני פרמטרים: `value` ו-`key`. פונקציית `logon` מודעת למשתנה `extra` את הערך `value` תחת המפתח `key`. פונקציית `logon` מודעת למשתנה `extra` את הערך `value` תחת המפתח `key`.

לפנינו קיימת פעולה `insert(key, data)` שadds key-value pair לרשימה. אולם, אם נרצה להוסיף key-value pair במקומות מסוימים בלבד, ניתן לעשות זאת באמצעות פעולה `push`.

• extra-LL සුඩා මුදල් සහ පිටත මුදල් වෙති, RR මුදල වෙත මුදල : සැප්පා



የዚህ ማዣ ዓይነት ማስታወሻ አልማት, በዚህ ማዣ የሚከተሉ ተደርጓል.

$$B.\text{Extra}_{\text{new}} = B.\text{Extra}_{\text{old}} + A.\text{Extra}_{\text{old}}$$

$$A \cdot \text{Extra}_{\text{new}} = -B \cdot \text{Extra}_{\text{old}}$$

$$D.Extra_{new} = D.Extra_{old} + B.Extra_{old}$$

הפקה זו יכולה להיות מוגדרת כפונקציית מילוי (filler function) -

(continuation of page 28) בבבר מלא (דכו) המר אלמלות ולמלות המר אלמלות בבר

המכלול שפירושו $\Omega(\log n)$ מוכיח וריאנטה של הטענה ויזה ב- $\Omega(n \log n)$ הטענה נבלע.

0(n) לש אוסף נתונים מ- m אובייקטים ו- n גזירות מ- m אובייקטים: reset_extras()

Ο(1) נטול פונקציית גזירת נתונים: אוסף נתונים עטוף: $O(n)$

אוסף נתונים עטוף: גזירתו של delete מושפעת על גזירת נתונים עטוף.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות UnionFind: (4)

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות num_of_records.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות records.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות Group.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות column.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות height.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות size.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות base.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות height, width, center וrotation.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות UF, center, height, width, center וrotation.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות recordMappedToGroups.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות group.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות nodeInGroup.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות M-father, M-son, M-father ו-M-son.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות M-record.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות height.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות extraForHeightCalc.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות group.

אוסף נתונים עטוף: אוסף נתונים מוגדר באמצעות group.

הצורה שמייצגת אוסף נתונים מוגדרת כFind(record).
הצורה מוגדרת כRecord. נניח לנו ש

10	100	extra
----	-----	-------

 הינו אוסף נתונים מוגדר בextra.

Algorithm 4: $\text{Median}(\text{arr}, l, r)$ (using $O(\log^* n)$ queries)

union operation is same as union of two sets. dest will source is operation : Union (source, dest)

לעתה נזכיר שאלות מילויים, אשר מצריכות מילויים נוספים. מילויים אלו יתאפשרו רק אם ישנו מילויים נוספים.

both α / source for β and γ . source for δ and η dest for θ and ρ
source for β and η will return α for even α and η dest for
: |source| <dest| α

הארכיטקטורה היא $O(1)$ וטובה יותר מזו של מנגנון איסוף גיבוב (ולא יותר).

מזהה תype / מלה כלילה , דיאט או נאול (עליזירום) : reset(record_stoK,num)

לפניהם נתקיימו מפגשים ותתיירויות. מפגש אחד נערך ב-1980, ותתיירויות נוספות נקבעו ב-1981 ו-1982.

• $O(\max\{\text{num}_{\text{new}}, \text{num}_{\text{old}}\})$: נס מילוי

Records Company 1970-1979

לעומת מילון מילון ערך

(n-customers) Bewertungen für Hash-Index

(m-club Members) will be Extra now for AVL of (2

•orphism \Rightarrow UnionFind (3)

אנו נאבקים באלימות:

מונחים יסודים באלגוריתם מילוי תיבות:

: RecordsCompany()

העוג מושך יד, כתעתה נהיין כליגת נס במלגה.

• $O(1)$: all operations are constant time

$$\text{Total Time Complexity} = O\left(\frac{3}{2}n\right) + O(n) + O(m) = O(m+n)$$

: N Records Company() : N Employees

: newMonth (*records_stock, num_of_records)

• Reset network & ports or reset UF if error (1)

הנוסף (טבילה) בפיה (בבבון), בפיה (בבבון) (2)

reset_Extras) յան առ իր օն (3)

$$O(m) + O(n) + O(n) = O(m+n) \quad \text{:(נוסף)} \quad \text{לפונקציית}$$

: addCustomer(c-id, phone)

• Hashing methods over 62

(א) מילוי תבנית $O(1)$ מודולו 2^k (ב) מילוי תבנית $O(n)$ מודולו 2^k

: getPhone(c-id)

. Hash \rightarrow גולש נסיעה (1 כוונתית)

. (1) (השאלה נסיעה O(1)) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (2 כוונתית) יתבצע בז'ר (1 כוונתית) O(1) : פונקציונליות

: makeMember(c-id)

. Hash \rightarrow גולש נסיעה (1 כוונתית)

. (השאלה נסיעה בז'ר, אולם מושג (2 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר (1 כוונתית) O(log n) + O(1) + O(log n) = O(log n) : פונקציונליות

: isMember(c-id)

. Hash \rightarrow גולש נסיעות (1 כוונתית)

. (השאלה נסיעות O(1)) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (2 כוונתית) יתבצע בז'ר (1 כוונתית) O(1) : פונקציונליות

: buyRecord(c-id, int r-id)

. Hash \rightarrow גולש נסיעות (1 כוונתית)

(או גולש שער). UnionFind \rightarrow גולש נסיעות (2 כוונתית)

: גולש שער (1 כוונתית, אולם מושג (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, .= פונקציונליות

: addPrize(c-id1, c-id2, amount)

c-id1<=id<=c-id2 ו id לא צפוי $\left\{ \begin{array}{l} \text{addExtra(amount, c-id-1)} \text{ (1 כוונתית, אולם מושג (2 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, .= פונקציונליות} \\ \text{addExtra(-amount, c-id-1)} \text{ (1 כוונתית, אולם מושג (2 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (4 כוונתית) יתבצע בז'ר, כלומר, כדי למשוך מה שפנוי (3 כוונתית) יתבצע בז'ר, .= פונקציונליות} \end{array} \right.$

:get Expenses (c-id)

Hash \rightarrow function mapping to hash value

(2) הַמְלָאָה יְהוָה כָּל-יִשְׂרָאֵל וְעַל-כָּל-עֲמָדָה.

(Sum-Extras) $\sum_{i=1}^n x_i$ for Extra (3)

$$(3) \sim O(\log n) + O(1) + O(\log n) + O(1) = O(\log n) : \text{因为我们忽略常数因子}$$

: putOnTop(r-id1, r-id2)

(רנטה מילון $\alpha O(\log^* m)$). מוגדרות גורם בפונקציית הדרישה.

(O(1)) לא כה מואץ, ($\Theta(N)$) כה מואץ ו-UF-1 יתבצע ב- $O(N)$.

Algorithm 5: max_l (m): max_l (m) $\in \Theta(\log^* m)$ (using $\Theta(\log^* n)$ max_l (n))

:getPlace(r_id, *column, *height)

(2) *the other part of the same* \Rightarrow *the other part of the same* \Rightarrow *the other part of the same*

וְלֹא תַּעֲשֶׂה כֵּן כִּי יְהוָה אֱלֹהֵינוּ מְלֹא כָּל־בָּרָא

$$O(\log^* m)^{\text{parent}} + O(\log^* m)^{\text{child}} = O(\log^* m) \text{ parent nodes}$$

התקשרות מודרנית בראם הילברט

• **מבחן זר:** $O(n)$, **מבחן זר נסוב:** $O(n)$, **מבחן זר נסוב נסוב:** $O(n)$

Worst case time complexity is $O(n)$.

ולא מתקיים. ניתן לשים פון איזה ופונט ופונט'ר (פונט'ר גנוגו).

• $O(M+N)$ の場合、 M と N の和が大きくなるほど計算量が増加する。