$$h = |E| \leq b^d - h$$

$$\text{Depth}(x, Q, D) \downarrow \text{if } \phi \text{ then NIL}$$

Depth

+ if h set

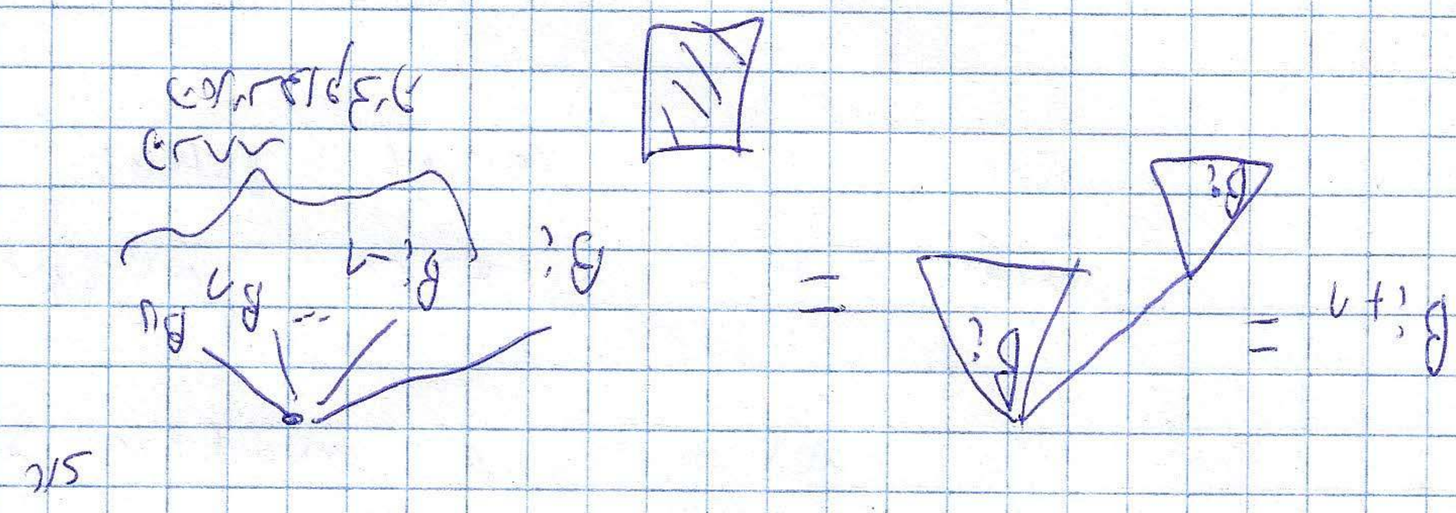
$$h = 1/1$$

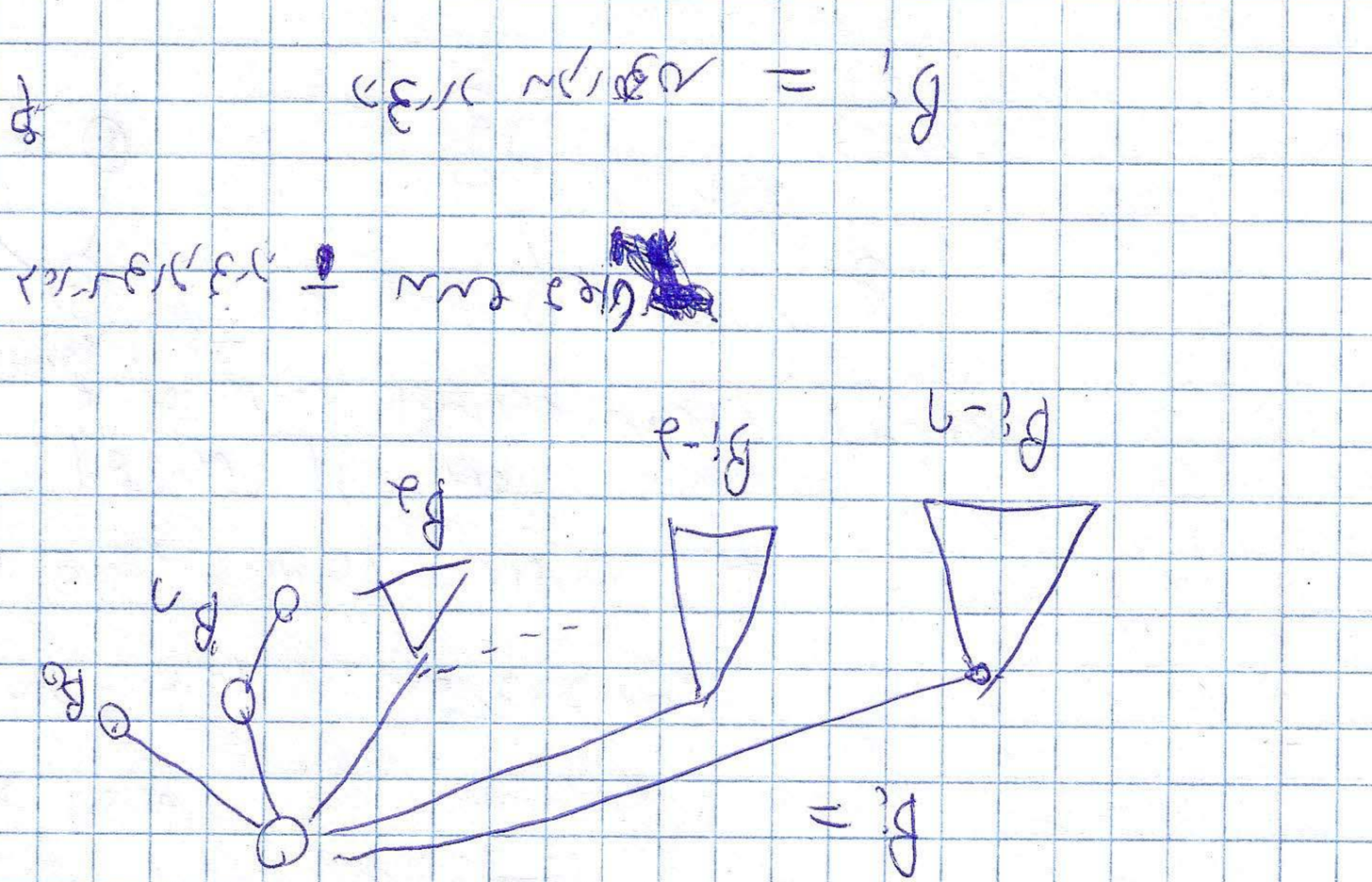
$$\text{Insert}(x, Q) \qquad h$$

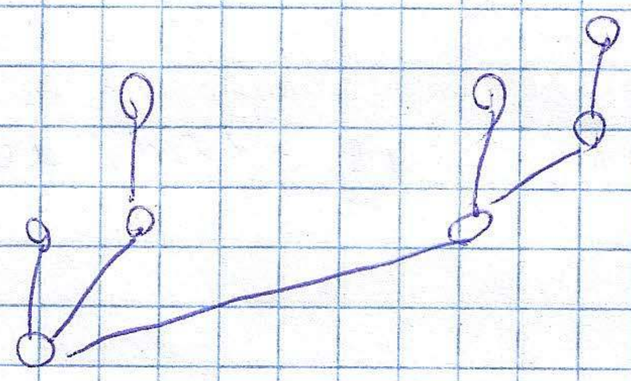
$$h.h(Q) \qquad h$$

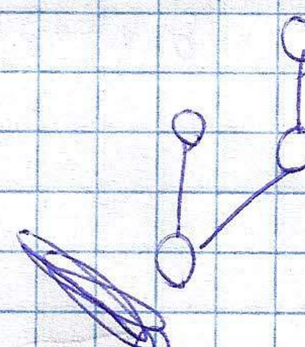
$$\text{del } h.h(Q) \qquad h$$

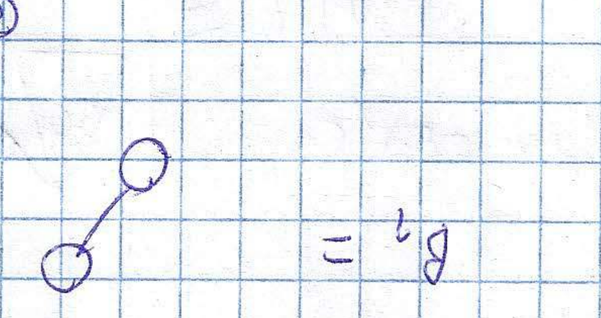
$$h \| \log h + h \cdot h$$

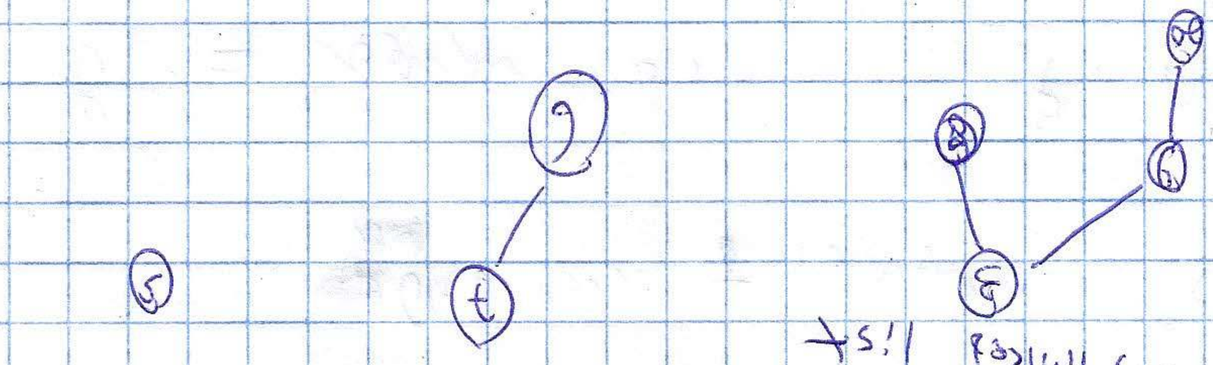


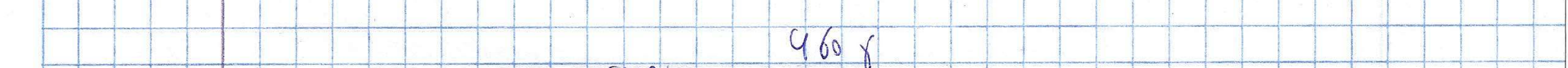












Allocated = (# Links + max_length) - (# Links_Max - max_length)

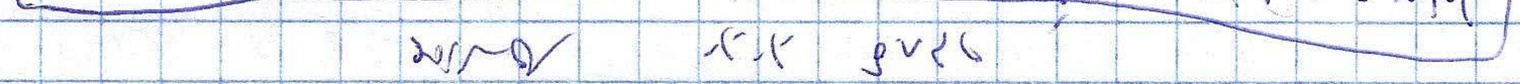
Already setup

Error link

Remove type

Add Login Node

Delete Node



Node Name

Node Type

Node Login

Node Password

Node Port

Node Status

Node Actions

Node Logs

Node Metrics

רק בעצים

Find min O(1) W.C.

Meld - O(1) W.C. → כאשר מוסיפים

Insert - O(1) W.C. → עצים חדשים

Lazy Meld

אלא אם רוצים להגביל [illegible] מספרים של צורך כניסתו בינוי,

ואז, כאשר סופרים כלשהו פולש [illegible] המינימום

של פוטנציאל $\Phi(c)$ = מס' העצים

delete min Amortized O(log)

הערימה פיבונאצ'י

נרצה לבצע כמה שיותר Decrease key

נרצה - Amortized O(1)

ש- amortized כי פיבונאצ'י. נרצה m גדולה מאוד.

זה מספיק?

מימוש Decrease key

אם x [illegible] נהיה קטן יותר מההורה שלו

נחתוך את העץ

של x ונוסיף אותו

לשורש ונעדכן את המינימום

אם צריך O(1) [illegible]

בעיה העץ כבר לא בינומי.

אולי נחשוב [illegible]. הבעיה הגובה [illegible]

לפי [illegible] עמוק עצים.

Collision Handling Methods

Open Addressing

Chaining

Perfect Hash Functions

Bloom Filters

Cuckoo Hashing

Rehashing

Extendible Hashing

Linear Probing

Robin Hood Hashing

Hopscotch Hashing

Table of Contents

Back to the Tree = Degree of the Root

Binomial Heaps - Fibonacci Heaps

Pairing Heaps - Brodal Heaps

Soft Heaps - Ranked Pairing Heaps

Interval Heaps - Min-max Heaps

Treaps - Randomized Binary Search Trees

Weight-balanced Trees - Scapegoat Trees

Red-black Trees - Splay Trees

AVL Trees - Tango Trees

2-3 Trees - 2-3-4 Trees

B-trees - B+ trees

Kd Trees - Interval Trees

Range Trees - Priority Search Trees

Segment Trees - Interval Trees

Quadtrees - Octrees

Tries - Suffix Trees

Suffix Arrays - Compressed Suffix Arrays

FM-index - Generalized Suffix Trees

B-K trees - Fusion Trees