Treap

Big O:

- space O(n)
- time
  - search worst O(n), average O(log(n))
  - insert worst O(n), average O(log(n))
  - delete worst O(n), average O(log(n))

Advantages:

- treap is same shape regardless of history
  - security: cant tell history
  - efficient sub tree sharing
  - useful for sets

Notes:

- heap invariant (children less or equal to parent)?
- formed by inserting nodes highest priority first into a BST without rebalancing
- each node has priority (heap) and key (BST)