Announcements/Reminders:

Last class:

• Balanced Search Trees

• Red-Black Trees
  • Concept
  • Properties
  • insert operation
  • Complexities

Today:

• RBT example

• Hashing
RBT insertion example

Starting with an empty RBT, insert 7, 14, 18, 23, 1, 11, 20, 29, 25, 27 (same values as before)
Complexity of RBT operations

print:

lookup:

insert:

delete:
Hashing

What problem are we solving?

Idea:

Terminology:

- Key
- Hash table
- Table Size (TS)
- Load Factor (LF)
- Hash function
- Collision
- Ideal hashing
Ideal hashing

Student records example:

```c
void insert(K key, D data)

D lookup (K key)

void delete(K key)
```

What if ID numbers are 10-digit numbers:

```
9012345789 9012345432 9023456789 ...```
Collision Handling (basic)
Choosing an appropriate table size

Resizing:

Steps:
Choosing a good hash function

Properties:

Two-step process:
Some Hashing Techniques

% tablesized

Extraction

Folding

Mid-square

Multiplication

Rotation/Weighting/Use-your-imagination/...
What if the keys are not integers?

Java's built-in hash function for Strings: