06 – Hash Tables

1. Hash Tables
    A. Use hash function to map keys and values
    B. Static : mod the key by array's size and map the value to that slot
    collision issue
    C. Design decision
        i. Hash function : speed / collision rate
        ii. Collision handling : large hash table / more instructions
2. Hash Functions
    A. Don't use cryptographic hash functions (too slow)
3. Static Hashing Schemes
    A. Linear Probing
        i. Non unique keys – separate chaining or redundant keys
    B. Robin Hood
        i. Variation of linear probing
        ii. Steal slots from rich keys(less jumps) to poor keys(more jumps)
    C. Cuckoo
        i. Use multiple hash table with various hash functions
        ii. Select one hash table that has empty slot, if not exists, evict one value and re-hash it

4. Dynamic Hashing Schemes
    A. Chained Hashing
        i. Hash table with linked list like bucket
    B. Extendible Hashing
        i. Similar to chained hashing
        ii. Bucket is also hash table, if collision occurs, use another hash function to make another hash table with keys that make collision
    C. Linear hashing
        i. Maintain bucket
        ii. If bucket is full, split it with new bucket (overflow page)
5. What is it?
    A. cryptographic hash functions