Lecture 15

Hashing

Heap Sort

- More swaps in general.
- Time complexity: O($n\log_2 n$)

Hashify

- Super fast data retrieval
- Hash table = Array (usually)
  - Unsorted = O(n) : Slow
  - Sorted = O($\log_2 n$) : Fast
  - Hash Table = O(1) : Superfast
- Retrieval using arrays is O(1)
- Arrays are Fixed/Rigid/Static data structure

Uses of Hashing

- World Wide Web [Google Search]
- Train/Flight booking systems, stock market websites [IP address]
- Search a word in a pdf file.
- Playing a live real-time multi-player game, possibly with high-end graphics
- Cryptographic password storage.
- Fingerprints
- Database indexing
- Caching systems

Methods of Hashing

- EVE = 5 + 22 + 5 = 32 = 3 + 2 = 5
- FEZ = 6 + 5 + 26 = 37 = 10 = 1

Create a function $f(x)$: Takes in a string and returns a number.

This function is called a *Hash function.*

Collisions

Situations where $X_1 \neq X_2$ but hashes are equivalent.

Load Factor

$$(\alpha) \text{ Load Factor} = \frac{\text{Number of Items Stored}}{\text{Size of Array}}$$