Further discussion of Hashing

Collision Resolution

What are your questions? About…

- SlidingBlocks
- WA12
- Recurrence Relations
- Quicksort
- Binary Heaps
- Hash Tables
- Anything Else

Quadratic probing

- With linear probing, if there is a collision at H, we try H, H+1, H+2, H+3,... until we find an empty spot.
  - Causes (primary) clustering
- With quadratic probing, we try H, $H+1^2$. $H+2^2$, $H+3^2$,...
  - Eliminates primary clustering, but can cause secondary clustering.

Hints for quadratic probing

- **Choose a prime number for the array size**
  - Guaranteed insertion and no cell is probed twice, provided That the table is no more than half full.
  - Suppose the array size is P, a prime number greater than 3
  - Show by contradiction that if i and j are $\leq \lfloor P/2 \rfloor$, and $i \neq j$, then $H + i^2 \pmod P \not\equiv H + j^2 \pmod P$.
- **Use an algebraic trick to calculate next index**
  - Replaces mod and general multiplication with subtraction and a bit shift
  - Difference between successive probes:
    - $H + (i+1)^2 = H + i^2 + (2i+1)$ [can use bit-shift for multiplication].
    - nextProbe = nextProbe + (2i+1);
      if (nextProbe >= P), nextProbe -= P;

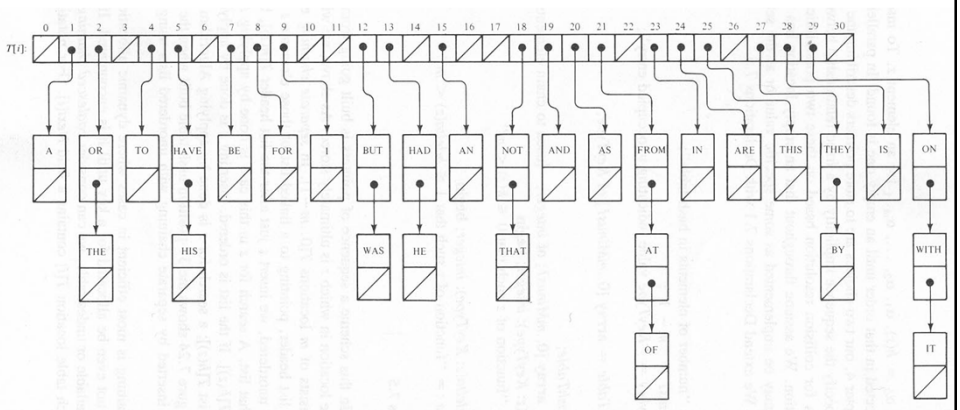
Quadratic probing analysis

- No one has been able to analyze it
- Experimental data shows that it works well
  - Provided that the array size is prime, and is the table is less than half full

Other approaches to collision resolution

- Double hashing
  - A second hash function is used to calculate an offset **d** to use in probing. Try locations h+d, h+2d, h+3d, etc
- Separate chaining
  - Rather than an array of items, we use an array of linked lists. When multiple items hash to the same location, we add them to the list for that location
  - Picture on next slide
  - No clustering effect
    - But we use space (that could have been used to make the array larger) for the links.
    - If many items have the same hash code, the chains can become long.

Hashing with Chaining



Hash Table Exercise

~40 minutes
On a handout and in your repository
Do it with your "scrabble team"
There's a handout for everyone, but only one submission per team

Data Compression

Fixed-length character codes
Variable-length character codes

Data compression

- How to fit data into a smaller space without losing any information?
- CPU calculations are many times faster than disk operations or network transmissions
- Thus it is faster to compress data before storing/sending it

Data Compression

YOU SAY GOODBYE. I SAY HELLO. HELLO, HELLO. I DON'T KNOW WHY YOU SAY GOODBYE, I SAY HELLO.

Letter frequencies

SPACE	17	A	4	U	2
O	12	S	4	W	2
Y	9	I	3	N	2
L	8	D	3	K	1
E	6	COMMA	2	T	1
H	5	B	2	APOSTROPHE	1
PERIOD	4	G	2		

- There are 90 characters altogether.
- How many total bits in the ASCII representation of this string?
- We can get by with fewer bits per character (custom code)
  - How many bits per character? How many for entire message?
  - Do we need to include anything else in the message?
  - How to represent the table?
    1. count
    2. ASCII code for each character How to do better?