PROBLEM SOLVING 5

Hashing, etc.

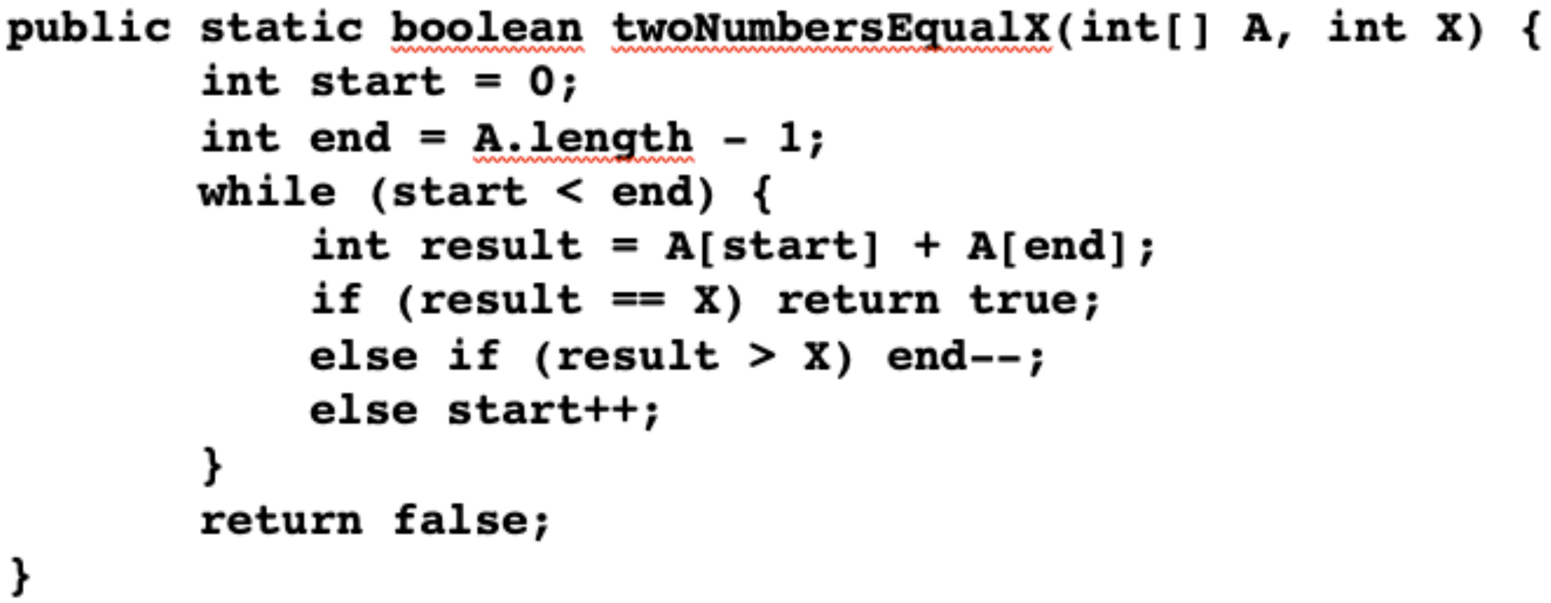
CS340

From Sorting to Hash Tables

- Can items be sorted by inserting into a hash table?
  - What data structures helped with sorting?
  - Can a hash table help under special conditions?
  - What are the issues?

Homework1

What was the brute force solution and time complexity?
What is the time complexity of the approach below?



```
public static boolean twoNumbersEqualX(int[] A, int X) {
        int start = 0;
        int end = A.length - 1;
        while (start < end) {
            int result = A[start] + A[end];
            if (result == X) return true;
            else if (result > X) end--;
            else start++;
        }
        return false;
}
```

CODING
INTERVIEW

Which are the benefits of a hash table?

1. Uses less memory than a binary search tree
2. Fast O(1) successor and minimum operations on average, like a tree
3. Fast O(1) search and insert operations on average
4. All of the above
5. None of the above

Interview Questions

- Professor Marley hypothesizes that he can obtain substantial performance gains by modifying the chaining scheme to keep each list in sorted order. How does the professor’s modification affect the running time for successful searches, unsuccessful searches, insertions, and deletions

Interview Questions

- Consider a hash table of size 7 with hash function:
  $h(k)= k \bmod 7$
- Insert in order 16, 23, 9, 14, 21
- What is result when collisions are handled by linear probing

Probing Techniques: Double hashing

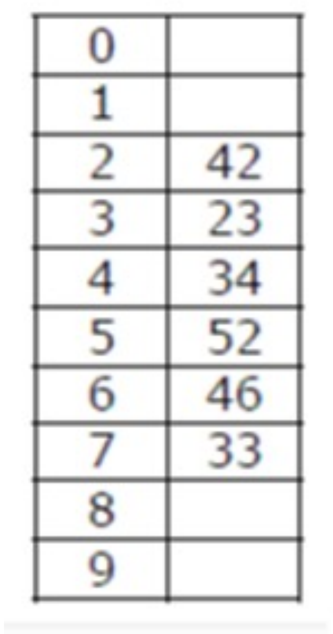
- Use two auxiliary hash functions, h1 and h2. h1 gives the initial probe, and h2 gives the remaining probes:
  $h(k, i) = (h1(k) + ih2(k)) \bmod m$.

Interview Questions

- Consider a hash table of size 7 with hash function:
  h1(k)= k mod 7
- Insert in order 16, 23, 9, 14, 21
- When collisions are handled by double hashing where the second hash function is h2(k) = 5 – (k mod 5).
- h(k, i) = (h1(k) + ih2(k)) mod m.

A hash table of length 10 uses open addressing with hash function h(k)=k % 10.

- The resulting table:



0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

- In what order were items inserted?

1. 46, 42, 34, 52, 23, 33
2. 34, 42, 23, 52, 33, 46
3. 46, 34, 42, 23, 52, 33
4. 42, 46, 33, 23, 34, 52
5. None of the above

H(x) = (3x +4) mod7. Linear probing. 1, 3, 8, 10 are inserted.

num	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
1.	8			3	1		10
2.	1	8	10				3
3.	1			10	8		3
4.	1	10	8				3

5. None of the above.

Input 4322, 1344, 1471, 9679, 1989, 6171, 6173, 4199 h(x)= x % 10. Which are true?

1. 9679, 1989, 4199 hash to the same value
2. 1471, 6171 hash to the same value
3. All elements hash to the same value
4. No elements hash to the same value

Project 3

- Experiments,
  - How do the numbers from part 1 change if words are added to the beginning of the linked list instead of to the end?
  - How does the number of steps change if the words are held in a binary search tree or red-black tree instead of a linked list? How many steps does theory predict*** as opposed to how many are actually required?
  - What happens if you change the hash function (write your own) and why?
  - If you choose another novel, are the most frequent words substantially the same?
  - Make up your own experiment