Q0, 1



#### Maximum Contiguous Subsequence Sum

After today's class you will be able to:

provide an example where an insightful algorithm can be much more efficient than a naive one.



Andrew Hettlinger ► Matt Boutell November 6 at 12:30pm · 44

In your class, I never thought I'd actually use big O notation, but now I find myself using it in my complaints to coworkers about how a previous developer would sort a list before doing a binary search to find a single element O(nlogn) + O(logn) instead of just doing a linear search O(n). I feel really nerdy now (as if I didn't before 🙂 )

 $\sim$ 

Like · Comment

### Recap: MCSS

*Problem definition*: Given a non-empty sequence of *n* (possibly negative) integers  $A_1, A_2, ..., A_n$ , find the maximum consecutive subsequence  $S_{i,j} = \sum_{k=i}^{j} A_k$ , and the corresponding values of *i* and *j*.

Recap: Eliminate the most obvious inefficiency, get  $\Theta(N^2)$ 

for( int i = 0; i < a.length; i++ ) {</pre> int thisSum = 0; for( int j = i; j < a.length; j++ ) {</pre> thisSum += a[ j ]; if( thisSum > maxSum ) { maxSum = thisSum;segStart = i; seqEnd = j; } ł

### MCSS is $O(n^2)$

- Is MCSS  $\theta(n^2)$ ?
  - Showing that a problem is Ω (g(n)) is much tougher. How do you prove that it is impossible to solve a problem more quickly than you already can?
  - Can we find a yet faster algorithm?

 $\begin{array}{l} f(n) \text{ is } O(g(n)) \text{ if } f(n) \leq cg(n) \text{ for all } n \geq n_0 \\ \circ \text{ So O gives an upper bound} \\ f(n) \text{ is } \Omega(g(n)) \text{ if } f(n) \geq cg(n) \text{ for all } n \geq n_0 \\ \circ \text{ So } \Omega \text{ gives a lower bound} \\ f(n) \text{ is } \theta(g(n)) \text{ if } c_1g(n) \leq f(n) \leq c_2g(n) \text{ for all } n \geq n_0 \\ \circ \text{ So } \theta \text{ gives a tight bound} \\ \circ \text{ f(n) is } \theta(g(n)) \text{ if it is both } O(g(n)) \text{ and } \Omega(g(n)) \end{array}$ 

# Maximum Contiguous Subsequence Sum

#### A linear algorithm.

 $\{-3, 4, 2, 1, -8, -6, 4, 5, -2\}$ 



#### **Q2**

### **Observations?**

▶ Consider {-3, 4, 2, 1, -8, -6, 4, 5, -2}



- Any subsequences you can safely ignore?
  - Discuss with another student (2 minutes)

### Observation 1

- We noted that a max-sum sequence A<sub>i,j</sub> cannot begin with a negative number.
- Generalizing this, it cannot begin with a prefix A<sub>i,k</sub> with k<j whose sum is negative.</li>
  - Proof by contradiction. Suppose that A<sub>i,j</sub> is a maxsum sequence and that S<sub>i,k</sub> is negative. In that case, a larger max-sum sequence can be created by removing A<sub>i,k</sub> However, this violates our assumption that A<sub>i,j</sub> is the largest max-sum sequence.

### Observation 2

- All contiguous subsequences that border the maximum contiguous subsequence must have negative or zero sums.
  - Proof by contradiction. Consider a contiguous subsequence that borders a maximum contiguous subsequence. Suppose it has a positive sum. We can then create a larger max-sum sequence by combining both sequences. This contradicts our assumption of having found a max-sum sequence.

### **Observation 3**

- No max-sum sequence can start from inside a subsequences that has a negative sum and extend beyond it.
- In other words, if we find that S<sub>i,j</sub> is negative, we can skip all sums that begin with any of A<sub>i</sub>, A<sub>i+1</sub>, ..., A<sub>j</sub>.
- We can "skip i ahead" to be j+1.

Observation 3 For any *i*, let  $j \ge i$  be the smallest number such that  $S_{i,j} < 0$ .

Then for any *p* and *q* such that  $i \le p \le j$  and  $p \le q$ :

- either  $A_{p,q}$  is not a MCS, or
- S<sub>p,q</sub> is less than or equal to a sum already seen (i.e., one with subscripts less than *i* and *j* respectively).

- Suppose q > j, then  $A_{i,j}$  is part of  $A_{i,q}$  and (by Obs. 1)  $A_{i,q}$  is not a MCS. But  $S_{i,q} \ge S_{p,q}$ , so  $A_{p,q}$  is not a MCS either.
- Suppose  $q \le j$ , then  $S_{i,q}$  is a "sum already seen". Since  $S_{p,q} \le S_{i,q}$  the claim holds.

• Suppose q > j, then  $A_{i,j}$  is part of  $A_{i,q}$  and (by Obs. 1)  $A_{i,q}$  is not a MCS. But  $S_{i,q} \ge S_{p,q}$ , so  $A_{p,q}$  is not a MCS either.



• Suppose q > j, then  $A_{i,j}$  is part of  $A_{i,q}$  and (by Obs. 1)  $A_{i,q}$  is not a MCS. But  $S_{i,q} \ge S_{p,q}$ , so  $A_{p,q}$  is not a MCS either.



• Suppose  $q \le j$ , then  $S_{i,q}$  is a "sum already seen". Since  $S_{p,q} \le S_{i,q}$  the claim holds.

#### Q5, Q6

### New, improved code!

```
public static Result mcssLinear(int[] seq) {
    Result result = new Result();
    result.sum = 0;
    int thisSum = 0;
    int i = 0;
    for (int j = 0; j < seq.length; j++) {</pre>
        thisSum += seq[j];
        if (thisSum > result.sum) {
            result.sum = thisSum;
            result.startIndex = i:
                                              S<sub>i.i</sub> is negative. So,
            result.endIndex = j;
        } else if (thisSum < 0) {</pre>
                                                skip ahead per
            // advances start to where end
                                                 Observation 3
            // will be on NEXT iteration
            i = i + 1;
            thisSum = 0;
        }
                          Running time is is O (?)
    return result;
                          How do we know?
}
```

### What have we shown?

- MCSS is O(n)!
- Is MCSS  $\Omega(n)$  and thus  $\theta(n)$ ?
  - Yes, intuitively: we must at least examine all n elements

### Time Trials!

- From SVN, checkout MCSSRaces
- Study code in MCSS.main()
- For each algorithm, how large a sequence can you process on your machine in less than 1 second?

## MCSS Conclusions

- The first algorithm we think of may be a lot worse than the best one for a problem
- Sometimes we need clever ideas to improve it
- Showing that the faster code is correct can require some serious thinking
- Programming is more about careful consideration than fast typing!

### Interlude

- If GM had kept up with technology like the computer industry has, we would all be driving \$25 cars that got 1000 miles to the gallon.
   Bill Gates
- If the automobile had followed the same development cycle as the computer, a Rolls-Royce would today cost \$100, get a million miles per gallon, and explode once a year, killing everyone inside.

- Robert X. Cringely

### Stacks and Queues

A preview of Abstract Data Types and Java Collections

This week's major program

Q7-8

### Stacks and Queues Part 1

An exercise in implementing your own growable circular Queue:

- 1. Grow it as needed (like day 1 exercise)
- 2. Wrap-around the array indices for more efficient dequeuing

Discuss Stacks as a warmup (push, pop), then ideas for Queues (enqueue, dequeue)

Analyze implementation choices for Queues – much more interesting than stacks!

### Stacks Part 2: Evaluator

An exercise in writing cool algorithms that evaluate mathematical expressions:

Infix: 6 + 7 \* 8 Postfix: 6 7 8 \* +

Both using stacks.

### Meet your partner

- Plan when you'll be working
- Review the pair programming video as needed
- Check out the code and read the specification together