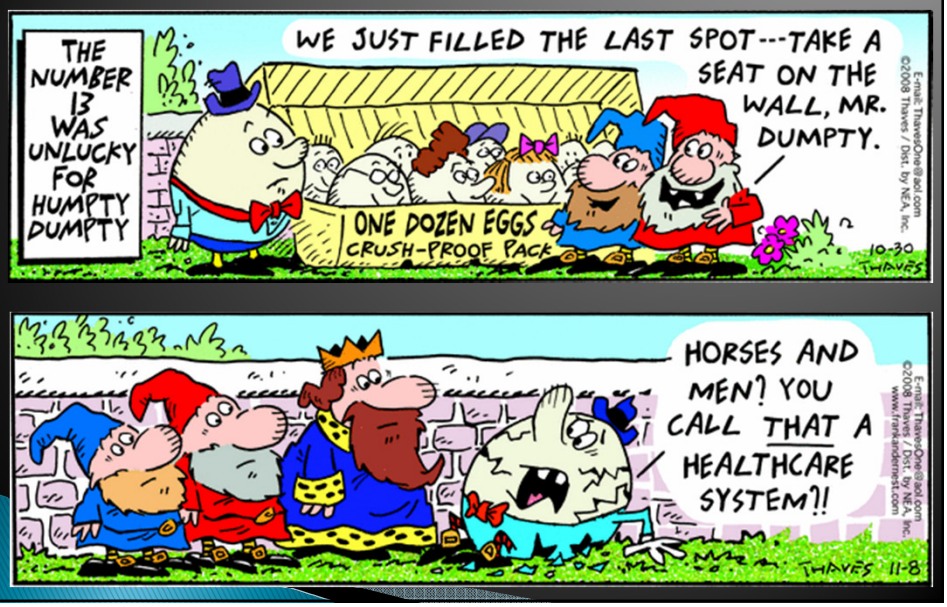
Q1

CSSE 230 Day 6

Linear Algorithm for MCSS

Check out from SVN: **MCsSRaces**
Answer Quiz Question 1

Questions?



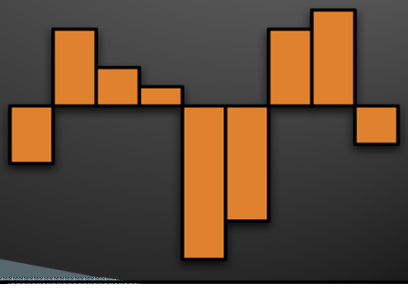
Agenda

- Maximum Contiguous Subsequence Sum Linear Algorithm
- Finite State Machines
  - Implementation strategies
- Work Time

Maximum Contiguous Subsequence Sum

- A linear algorithm.

{−3, 4, 2, 1, −8, −6, 4, 5, −2}



Q1, if you haven't yet

Recap: MCSS

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

- In {−2, **11, −4, 13**, −5, 2}, MCSS is $S_{2,4}$ = ?
- In {1, −3, 4, −2, −1, 6}, what is MCSS?

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

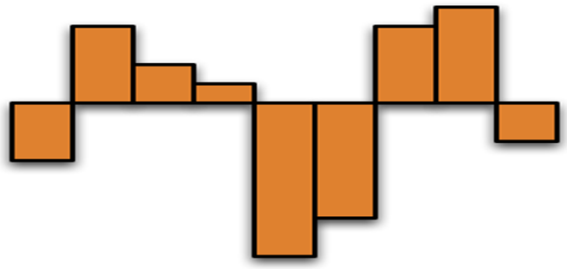
```
for( int i = 0; i < a.length; i++ ) {
    int thisSum = 0;
    for( int j = i; j < a.length; j++ ) {
        thisSum += a[ j ];

        if( thisSum > maxSum ) {
            maxSum = thisSum;
            seqStart = i;
            seqEnd   = j;
        }
    }
}
```

We can do even better than this!

Q2

Observations?

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



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

Q3

Observation 1

- We noted that a max-sum sequence $A_{i,j}$ cannot begin with a negative number.
- Generalizing this, it cannot begin with a prefix ($A_{i,k}$ with $k<j$) whose sum is negative.
  - **Proof: If $S_{i,k}$ is negative, then $S_{k+1,j} > S_{i,j}$, so $A_{i,j}$ would not be a sequence that produces the maximum sum.**

Observation 2

- All contiguous subsequences that border the maximum contiguous subsequence must have negative (or zero) sums.
  - **Proof**: If one of them had a positive sum, we could simply append (or "prepend") it to get a sum that is larger than the maximum. Impossible!

Q4-5

Observation 3

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

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

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

So What!?

- If we find that $S_{i,j}$ is negative, we can skip all sums that begin with any of A_i, A_{i+1}, …, A_j.
- There is no new MCS that starts anywhere between A_i and A_j.
- So we can "skip i ahead" to be j+1.

Observation 3 again:

> For any i, let $j \geq i$ be the smallest number such that $S_{i,j} < 0$.
>
> Then for any p and q such that $i \leq p \leq j$ and $p \leq q$:
>
> - either $A_{p,q}$ is not a MCS, or
> - $S_{p,q}$ is less than or equal to a sum already seen (i.e., one with subscripts less than i and j respectively).

New, improved code!

Q6

```
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++) {
        thisSum += seq[j];

        if (thisSum > result.sum) {
            result.sum = thisSum;
            result.startIndex = i;
            result.endIndex = j;
        } else if (thisSum < 0) {
            // advances start to where end
            // will be on NEXT iteration
            i = j + 1;
            thisSum = 0;
        }
    }
    return result;
}
```

$S_{i,j}$ is negative. So, skip ahead per Observation 3

Running time is is Θ (?)
How do we know?

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!

More About Colorize

- FSM representations

Partners

Possible Representations of the Finite State Machine

Q7

Diagrams on the whiteboard

- 2-Dimensional array:
  - Rows indexed by state, Columns by input character.
  - Each array entry is a pair object (as in DS Section 3.7):
    - [next state, what to print]
- Monolithic controller with nested switch statements
- The first choice may be more efficient and have shorter code
- The second choice is probably easier to write and modify
  - Can be made more modular by having a method for each state

Other issues that you may need to investigate for the Colorize program

- The java Keywords
  - http://java.sun.com/docs/books/tutorial/java/nutsandbolts/_keywords.html
- How to open files for reading and writing, plus reading and writing a line at a time:
- Some basic HTML
- There are descriptions and/or links for all of these topics in the assignment document

Good code (all programming assignments)

- **Good comments**:
  - Javadoc comments for public fields and methods.
  - Explanations of anything else that is not obvious.
- **Good variable and method names**:
  - Eclipse has name completion (ALT /), so the "typing cost" of using long names is small
- **Use local variables and static methods** (instead of fields and non-static methods) where appropriate
  - "where appropriate" includes any place where you can't explicitly justify creating instance fields
- **Consistent indentation** (ctrl-shift f)
- **No super-long lines of code**
- **Blank lines between methods, space after punctuation**

Hardy/Colorize repos : Section 01

csse230-201230-hardy-11,amesen,piliseal
csse230-201230-hardy-12,dingx,elswicwj
csse230-201230-hardy-13,eubankct,murphysw
csse230-201230-hardy-14,goldthea,postcn
csse230-201230-hardy-15,huangz,namdw
csse230-201230-hardy-16,maglioms,mcdonabj
csse230-201230-hardy-17,mccullwc,yuhasmj
csse230-201230-hardy-18,mehrinla,newmansr
csse230-201230-hardy-19,millerns,timaeudg
csse230-201230-hardy-20,morrista,rudichza,koestedj
csse230-201230-hardy-21,nuanests,shahdk
csse230-201230-hardy-22,rujirasl,semmeln
csse230-201230-hardy-23,sanderej,weirjm
csse230-201230-hardy-24,jarvisnw,harbisjs
csse230-201230-hardy-25,paulbi,woolleld

Hardy/Colorize repos : Section 02

csse230-201230-hardy-26,bollivbd,memeriaj
csse230-201230-hardy-27,davelldf,toorha
csse230-201230-hardy-28,ewertbe,hopwoocp
csse230-201230-hardy-29,faulknks,spryct,scroggd
csse230-201230-hardy-30,fendrirj,pohltm
csse230-201230-hardy-31,gartzkds,mengx
csse230-201230-hardy-32,haydr,lawrener
csse230-201230-hardy-33,lius,weil
csse230-201230-hardy-34,minardar,watterlm
csse230-201230-hardy-35,modivr,roetkefj
csse230-201230-hardy-36,qinz,zhangz
csse230-201230-hardy-37,stewarzt,uphusar
csse230-201230-hardy-38,ruthat,tilleraj
csse230-201230-hardy-39,iwemamj,meyermc
csse230-201230-hardy-40,taylorem,yuhasem

Finish and turn in the quiz

- Work time:

Pascal
HardyPart2
Colorize