Algorithms and OOD (CSC 207 2014S) : EBoards

CSC207.01 2014S, Class 48: Dynamic Programming

Overview

- Preliminaries.
  - Admin.
  - Questions.
- The stamps problem.
- Fibonacci.
- Generalizing the idea.
- The stamps problem, revisited.
- Edit distance in strings. [Maybe]

Preliminaries

Admin

- Don't forget to turn in tutor/mentor/grader forms.
- What food/drink do you want on Monday?
  - EW wants grapefruit juice
  - Others want chocolate milk
  - Fruit - Blueberries, Other berries, Bananas, Grapes

Upcoming Work

- Reading for Monday: No reading. Prepare your talks.
  - Get me slides by 8pm Sunday night.
- Today's writeup: No writeup.
- Part 2 of the project is due tonight at 10:30 p.m. (No, I'm not updating the Web page.)
  - Please be careful on formatting!
  - Please be careful on citations!

Extra Credit

- CS Table today: Heartbleed.
- Iowater project April 26 - Tag drains. Mail iowater@grinnell.edu for details.
- Field day!
- ISO Cultural Show tonight at 7pm
- Titular Head Saturday night

Questions

- Essays can be a few paragraphs. (One on feature, one on license.)
- Documentation should be longer.

The stamps problem

- A famous optimization problem
- Set of values: s1, s2, s3, ... s_n (1, 2, 7, 12, 25)
    - You may assume that it's sorted.
- Target value: 68
- Question: Find the combination of values that makes the target value that has the fewest total values
- Example: Goal is 36 12, 12, 12 - three values 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 - thirty six ones 25, 7, 2, 2 - four values
- Note: Sometimes the problem is not solvable.

Question: How do you solve this?

Greed is Good

- Take as many of the largest value as possible
- Then take as many of the next largest value as possible
- ...
- Whoops ... Doesn't work 36 : 12, 12, 12, not 25, 7, 2, 2

EM

- Make every combination of values that create the target
- Figure out which requires the fewest items
- Slow, but probably correct.
- Question: How do you make all of the combinations? For s = s0 to sn compute the smallest set for (t-s) add s
- Can we make this faster? Yes. Memoize.

UM

- For l = 1 to ...
    - Make every combination of length l
    - If any of them equals the target, you're done
- Note: If there are k values required for a number, this is approximately O(n^k).

Fibonacci

- fib(0) = 0
- fib(1) = 1
- fib(n) = fib(n-1) + fib(n-2)

Why do we care? (other than that it’s cooly recursive)

- Breeding patterns of rabbits
- Growth rate of snails
- Gives golden ratio

Direct translation

```
public static BigInteger fib(int n)
{
  if (n < 2)
    return new BigInteger(n);
  else
    return fib(n-1).add(fib(n-2));
} // fib(int)
```

- This is insanely slow.

- Solution: Keep track of the past values

```
BigInteger FIB[];        // Cached results.  If a value in here is non-null
                         // it’s the nth Fibonacci number
public static BigInteger fib(int n)
{
  if (FIB == null)
    {
      FIB = new BigInteger[n+1];
      FIB[0] = 0;
      FIB[1] = 1;
    } // if we don’t have the array
  if (FIB[n] != null)
    return FIB[n];
  else
    {
      FIB[n] = fib(n-1).add(fib(n-2));
      return FIB[n];
    } // else
```

- Caching has turned an exponential algortihm into a linear algorithm

- We could also use the closed form of the Fibonacci numbers.

  - But that requires us to know math.

- We could also build the array iteratively

  ```
  public static BigInteger fib(int n)
  {
    BigInteger FIB[n+1];
    FIB[0] = 0;
    FIB[1] = 1;
    for (int i = 2; i <= n; i++)
      FIB[i] = FIB[i-1].add(FIB[i-2]);
    return FIB[n];
  } // fib(int)
  ```

- Sam thinks of the key ideas of dynamic programming as: "Cache in an array, and build from bottom up."

The stamps problem, revisited

Cache and build from bottom up

Reminder:

- n: The number of original values
- t: The target value

Running time: O(t*n) - Essentially linear.

Edit distance in strings

Nope, we didn’t get this far.