CS2223
HW#7

DUE: Thursday, April 27

1. (12 points) ELECTORAL COLLEGE TIES
The President of the United States is elected according to the Electoral College, where 568 votes are assigned to the candidates. We assume the following simplifications:

- each of the 50 states must cast **all** of its votes for the same candidate, and,
- there are two candidates, and,
- no write-in votes are allowed.

In reality these simplifications are not true; Maine and Nebraska can split their votes. The current electoral college votes for each state may be found at several sites on the Web, such as http://www.fec.gov/pages/elecvote.htm

Describe an efficient algorithm to test if a tie vote is possible under our simplified system. Every ten years the number of Electoral College votes a state receives can change, and the number of states could change, so simply solving the problem by hand for 2006 is not an acceptable solution. We want an efficient algorithm (polynomial in the number of states) to test if a tie is possible. Program your algorithm and execute the program to find a scenerio (a voting pattern) in which a two way tie could occur in 2008.

2. (8 points) Give an algorithm to compute $f(n)$, $n \in \mathbb{Z}^+$, the number of ways to make change for n¢ in the American system. We assume coins of values 1¢, 5¢, 10¢, 25¢, 50¢ and 100¢. Initial values of $f(n)$ are:

n	1	2	3	4	5	6	7
$f(n)$	1	1	1	1	2	2	2

3. (8 points) Do **Exercise *15.4-5*** on page 356 of our text.

CS2223
HW#7 SOLUTIONS

1. We can use the dynamic programming algorithm to solve the KNAPSACK PROBLEM to solve our current problem. The number of elements to be considered for packing, *n*, is the number of states. For each state *i*, its value v_i **and** its weight w_i are both the number of Electoral College votes the state has. The capacity of the Knapsack, *W*, is one half of the total number of Electoral College votes. A tie in the Electoral College corresponds to a packing of value *W*, so a tie is possible if and only if there is a packing of value *W*.

2. Let $g(n,m)$ denote the number of ways to make change for *n*¢ where the highest coin that could be used is an *m*¢ piece. So $f(n) = g(n,100)$. For $m \in \{1, 5, 10, 25, 50, 100\}$, we let partial function $p(m)$ denote the next smaller coiu. That is, $p(25) = 10$. If order doesn't matter, then a recurrence is

$$g(n,m) = \begin{cases} 0, \text{ if } n < 0 \\ 1, \text{ if } n \ge 0 \text{ and } m = 1 \\ 1 + g(n, p(m)), \text{ if } m = n > 1 \\ g(n, p(m)), \text{ if } m > n \\ g(n-m, m) + g(n, p(m)), \text{ if } n \ge 0 \text{ and } m > 1 \end{cases}$$

The two terms in the sum $g(n-m,m) + g(n, p(m))$ refer to using the *m*¢ piece at least once and to not using the *m*¢ piece, respectively.

```
(define (p m)
  (cond ((= m 100) 50)
        ((= m 50)  25)
        ((= m 25)  10)
        ((= m 10)   5)
        ((= m 5)    1)))
> (define (f n) (g n 100))
> (define (g n m)
    (cond ((< n 0) 0)
          ((and (>= n 0) (= m 1)) 1)
          ((and (= m n) (> n 1)) (+ 1 (g n (p n))))
          ((> m n) (g n (p m)))
          ((and (>= n 0) (> m 1)) (+ (g (- n m) m) (g n (p m))))))
> (f 10)
4
```

3. For sequence $\langle x_1, ..., x_n \rangle$, we compute $f(i)$, $1 \le i \le n$, which is the longest monotonically increasing sequence in $\langle x_1, ..., x_i \rangle$ which **must** include x_i. If this sequence doesn't include any member of $\langle x_1, ..., x_{i-1} \rangle$, then it has length 1. Otherwise, it is $1 + f(j)$ for some $1 \le j < i$ such that $x_j < x_i$.

$$f(i) = \max_{\substack{1 \le j < i \\ x_j < x_i}} \left(1, f(j) + 1\right)$$

The length of a longest monotonically increasing subsequence is $\max_{1 \le i \le n}\left(f(i)\right)$, and if i^* is the value of i which realizes the maximum, then the longest monotonically increasing subsequence contains x_{i^*}. One works backwards from i^* to construct the rest of the sequence.