CSE 565	**Martin Fürer**	**Algorithm Design and Analysis**

Assignment 8 Given: 10/18/18 Due: Wednesday 10/24/18

Exercises

Exercises are for your own practice. Don't hand them in. A version of them could show up in an exam though.

1. Design an algorithm to find a maximum independent set in the $2 \times n$ grid graph $G = (V, E)$. ($V = \{(i,j) : i \in \{1,2\} \text{ and } j \in \{1,\ldots,n\}\}$, $E = \{\{(i,j),(i',j')\} : |i-i'| + |j-j'| = 1\}$.)
2. Solve Exercise 6, Chapter 6 on page 317 of the Textbook.
(Pretty printing)

Problems

Problem solutions have to be handed in. A subset of them will be graded.

1. [5+5+10=20 points] A sequence of integers $a_1, \ldots, a_n$ is defined by $a_0 = 0$, $a_1 = 1$, and $a_j = 2a_{j-2} + a_{j-1}$ for $j \geq 2$.
   (a) Why would a recursive computation of a_n take $\Omega(2^{n/2})$ steps.
   (b) What is the running time using memoization, and the running time using dynamic programming?
   (c) Prove that $a_n = \Theta(2^n)$.
2. [5+5+5+5+5+5 = 30 points] (Knapsack)
In the Knapsack problem as we have seen it in class, we are given n items, where item i has a weight w_i and a value v_i. Furthermore, we have a capacity W of the knapsack. The goal is to pack as much value as possible into the knapsack. Assume now that we want the knapsack to be completely full, i.e., the total weight has to be exactly W. Among all full knapsacks, we still want to maximize the value. Let us call this new problem the Exact Knapsack problem. Think of modifying the Knapsack algorithm to obtain an algorithm for the Exact Knapsack problem.
Hint: When there is no possibility of producing a certain value w using some of the first i items, it is convenient to define $M[i, w] = -\infty$.

(a) Give a definition (in words) of the new $M[i, w]$ in order to solve the Exact Knapsack problem.

(b) Define the new values of base case $M[0, w]$.

(c) Write a recurrence equation for the new $M[i, w]$ with $i \geq 1$.

(d) What is the running time of the new algorithm for Exact Knapsack?

(e) Which of (b), (c), (d) have changed for the new algorithm, compared to the algorithm for the standard Knapsack?

(f) $M[n, W]$ is obviously the desired result of the new algorithm. Now assume, you have computed the whole M-table for Exact Knapsack, and you also want to know the result for the standard Knapsack. How can you get it from the computed M-table?

3. [5+10+10=25 points] (Subset Sum)
The special case of the Exact Knapsack problem with $w_i = v_i$ is known as the Subset Sum problem. In other words, we are given a sequence of positive integers $w_1, \ldots, w_n$ and a number W. We want to decide whether there is a subset of the w_i adding up to exactly W. (Here, we use this usual definition of "Subset Sum," even though our textbook uses term "Subset Sum" for the corresponding special case of the standard Knapsack problem.)

   (a) Write the recurrence equation (including the base case) for $M[i, w]$ defined as a boolean array.

   (b) Assume, you want to solve the counting problem associated with the Subset Sum problem, i.e., you want to compute the number of subsets S of $\{1, \ldots, n\}$ such that

$$\sum_{i \in S} w_i = W.$$

   Write pseudo-code for solving this counting problem.

   (c) Prove that your algorithm is correct by (strong) induction.