Problem 1
Consider the following function computing the Fibonacci number

Algorithm 1 FIBONACCI

function FIB(n)
  if $i == 0$ or $i == 1$ **then**
    return i
  else
    return Fib($i-1$) + Fib($i-2$);
  end if
end function

Draw the tree illustrating the recursive calls for FIBONACCI(5). It turns out that the number of leafs in a call tree for FIBONACCI is equal to the Fibonacci number itself, which is roughly 1.618^n. Write an iterative implementation of FIBONACCI(n) that only uses $O(n)$ time and space.

Problem 2
Show all intermediate steps of the dynamic programming algorithm for the weighted interval scheduling problem, for the following input.

item	1	2	3	4	5	6	7	8	9	10
start	0	1	0	3	2	4	6	2	7	6
finish	2	3	4	5	6	7	8	9	10	11
weight	2	9	6	5	7	11	8	10	4	6

Problem 3
Consider the following variant of the Interval Scheduling problem we saw in class. The input is defined by a set of intervals $I_1, \ldots, I_n$. We say that interval $I_i = (s_i, f_i)$ has length $f_i - s_i$. We would like to pick a set of non-intersecting intervals that use as much as possible of the common resource; that is, we want to maximize the sum of the lengths of the scheduled intervals. Design an efficient algorithm for this problem using dynamic programming.

Problem 4
Solve the following instance of the $\{0, 1\}$ Knapsack Problem with four items where the maximum allowed weight is $W_{\max} = 10$.

i	1	2	3	4
b_i	25	15	20	36
w_i	7	2	3	6

Problem 5
Suppose we are going on a hiking trip along the Great North Walk from Sydney to Newcastle. We have a list of all possible campsites along the way, say there are n possible places where we could camp. (Assume campsite are right off the path.) We want to do this trip in exactly k days, stopping in $k-1$ campsites to spend the night. Our goal is to plan this trip so that we minimize the maximum amount of walking done on

any one day. In other words, if our trip involves 3 days of walking, and we walk 11, 14, and 12 kilometers on each day respectively, the cost of this trip is 14. Another schedule that involves walking 11, 13, and 13 kilometers on each day has cost 13, and is thus preferable. The locations of the campsites are specified in advance, and we can only camp at a campsite.

Using dynamic programming, design an efficient algorithm for solving this problem. Your algorithm should run in $O(n^2k)$ time.

Problem 6

Consider a post office that sells stamps in three different denominations, 1c, 7c, and 10c. Design a dynamic programming algorithm that will find the minimum number of stamps necessary for a postage value of n cents. (Note that a greedy type of algorithm wont necessarily give you the correct answer, and you should be able to find an example to show that such a greedy algorithm doesnt work.) What is the running time of your algorithm?

Problem 7

Let $T(n)$ be the number of different binary search trees on n keys. For example, $T(1) = 1$ and $T(2) = 2$. Come up with a recurrence for $T(n)$. Use dynamic programming to compute $T(n)$ for a given n. Run some experiments to determine the asymptotic growth of $T(n)$.