Chapter 1

—–Material covered in class—–

1. Show the trace of Insertion-Sort on the following inputs.
   (a) $\{1, 7, 2, 6, 3, 5, 4\}$
   (b) $\{1, 2, 8, 7, 3, 4, 6, 5\}$
2. Identify a barometer instruction in Insertion-Sort.
3. Show the trace of Selection-Sort on the following inputs.
   (a) $\{1, 7, 2, 6, 3, 5, 4\}$
   (b) $\{1, 2, 8, 7, 3, 4, 6, 5\}$
4. Find a "best-case" input and a "worst-case" input for Selection-Sort.
5. Using the definitions of O, Ω and Θ, prove that $f(n) = 4n^2 + 9n^3 = \Theta(n^3)$.
6. Using the definitions of O, Ω and Θ, prove that $6n^2 + 20n = O(n^3)$ and $6n^2 + 20n \neq \Omega(n^3)$.
7. Using the limit criterion, prove that $2018n^4 + 4n^{2018} = \Theta(n^{2018})$.

—–Design & analysis of algorithms—–

8. Write the pseudo-code for Selection-Sort and analyze its running time in the worst case. How would you describe Selection-Sort in your own words?
9. Write an algorithm that finds the largest number in a list of n numbers and analyze its running time in the worst case. You should be able to write the pseudo-code of your algorithm. You should also be able to describe it in your own words.
10. (a) Write an algorithm that finds the m smallest numbers in a list of n numbers (where $1 \leq m \leq n$) and analyze its running time in the worst case. You should be able to write the pseudo-code of your algorithm. You should also be able to describe it in your own words.

(b) Can you do it in $O(n)$ time if $m = n - 2$?

11. Write an algorithm that finds both the smallest and largest numbers in a list of n numbers. Your algorithm must do at most $\frac{3}{2}n$ comparisons. You should be able to write the pseudo-code of your algorithm. You should also be able to describe it in your own words.

12. When is sequential search more appropriate than binary search?

13. (a) Write a $\Theta(n)$ time algorithm that sorts a list of n distinct integers, where each integer is in the interval $[1, 2019n]$. You should be able to write the pseudo-code of your algorithm. You should also be able to describe it in your own words.

    (b) Let k be a fixed positive integer. Write a $\Theta(n)$ time algorithm that sorts a list of n distinct integers, where each integer is in the interval $[1, kn]$. You should be able to write the pseudo-code of your algorithm. You should also be able to describe it in your own words.

    (c) What if the numbers are not necessarily distinct?

    (d) What if there is no upper bound on the numbers to be sorted?

14. What is the running time of the following algorithm?

    Input: The number n is even.

    for $i = 1$ to $\frac{3}{2}n$ **do**
        Print i
    end for
    for $i = n$ downto 1 **do**
        Print i
    end for

15. What is the running time of the following algorithm?

    Input: The number n is even.

    $j = 1$
    while $j \leq \frac{1}{2}n$ **do**
        $i = 1$
        while $i \leq j$ **do**
            Print (j, i)
            $i{+}{+}$
        end while
        $j{+}{+}$
    end while

16. What is the running time of the following algorithm?

    Input: The number n is divisible by 4.

    for $i = 2$ to n **do**

```
    for j = 0 to n do
        Print (i, j)
        j = j + ⌊n/4⌋
    end for
end for
```

17. What is the running time of the following algorithm?

    Input: The number n is a power of 2.

```
for i = 1 to n do
    j = n
    while j ≥ 1 do
        Print (i, j)
        j = ⌊j/2⌋
    end while
end for
```

18. What is the running time of the following algorithm?

    Input: The number n is a power of 2.

```
i = n
while i ≥ 1 do
    j = 1
    while j ≤ n do
        Print (i, j)
        j = 2 · j
    end while
    i = ⌊i/2⌋
end while
```

——Proofs——

19. Prove that $F_n \geq 2^{n/2}$ for all $n \geq 6$, where F_n is the n-th Fibonacci number. Why is this inequality false when $n < 6$?

20. Let

$$T(n) = \begin{cases} 2 & \text{if } n = 0, \\ 2 & \text{if } n = 1, \\ T(n-1) + T(n-2) + 5 & \text{if } n \geq 2. \end{cases}$$

    Prove that $T(n) \geq F_n$, where F_n is the n-th Fibonacci number.

21. Prove that $n \leq 2F_n$ for all $n \geq 0$.

22. Prove that $F_n < 2^n$ for all $n \geq 0$.

23. * Prove that the algorithm $fib(n)$ we studied in class makes a number of bit-operations that is proportional to $n \cdot F_n$, where F_n is the n-th Fibonacci number.
Note: you will need some of the properties you proved in the previous exercises.

24. Prove that

$$F_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

for all integers $n \geq 0$, where F_n is the n-th Fibonacci number.

25. Prove or disprove: $2^n = \Theta(3^n)$.

26. Group the following functions by complexity category and prove that your answer is correct.

    (a) $n\log(n)$ $\quad n^8$ $\quad n^{1+1/10}$ $\quad \left(1+\frac{1}{10}\right)^n$ $\quad \frac{n^2}{\log(n)}$ $\quad (n^2-n+1)^4$

    (b) $n!$ $\quad (n+1)!$ $\quad 2^n$ $\quad 2^{n+1}$ $\quad 2^{2n}$ $\quad n^n$ $\quad n^{\sqrt{n}}$ $\quad n^{\log(n)}$