CS170 Discussion Section 1: *1/18*

1. Asymptotic notation

(a) For each pair of functions $f(n)$ and $g(n)$, state whether $f(n) \in O(g(n))$, $f(n) \in \Omega(g(n))$, or $f(n) \in \Theta(g(n))$. For example, for $f(n) = n^2$ and $g(n) = 2n^2 - n + 3$, write $f(n) \in \Theta(g(n))$.

- $f(n) = n$ and $g(n) = n^2 - n$
- $f(n) = n^2$ and $g(n) = n^2 + n$
- $f(n) = 8n$ and $g(n) = n \log n$
- $f(n) = 2^n$ and $g(n) = n^2$
- $f(n) = 3^n$ and $g(n) = 2^{2n}$

(b) For each of the following, state the order of growth using Θ notation, e.g. $f(n) \in \Theta(n)$.

- $f(n) = 50$
- $f(n) = n^2 - 2n + 3$
- $f(n) = n + \cdots + 2 + 1$
- $f(n) = n^{100} + 1.01^n$
- $f(n) = n^{1.1} + n \log n$

2. Analyze the running time

For each pseudo-code snippet below, give the asymptotic running time in Θ notation. Assume that basic arithmetic operations ($+$,$-$,$\times$, and $/$) are constant time.

(a)
```
for i := 1 to n do
    j := 0;
    while j ≤ i do
        j := j + 2
```

(b)
```
s := 0;
i := n;
while i ≥ 1 do
    i := i div 2;
    for j := 1 to i do
        s := s + 1
```

(c)
```
i := 2;
while i ≤ n do
    i := i^2
```

(d)
```
for i := 1 to n do
    j := i^2;
    while j ≤ n do
        j := j + 1
```

3. Four-part Algorithm Practice

Given a sorted array A of n integers, you want to find the index at which a given integer k occurs, i.e. index i for which $A[i] = k$. Design an efficient algorithm to find this i.

Main idea:

Psuedocode:

Proof of correctness:

Running time analysis:

4. Sorted Array

Given a sorted array A of n (possibly negative) distinct integers, you want to find out whether there is an index i for which $A[i] = i$. Give a divide-and-conquer algorithm that runs in time $O(\log n)$.

5. Computing Factorials

Consider the problem of computing $N! = 1 \times 2 \times \cdots \times N$.

(a) If N is an n-bit number, how many bits long is $N!$, approximately (in $\Theta(\cdot)$form)?

(b) Give an algorithm to compute $N!$ and analyze its running time.