1. a) for all n, $f(n)=\Omega(g(n))$, c=1
$n\geq 300$, $f(n) = O(g(n))$, c=2
$f(n) = \Theta(g(n))$

b) for all n, $n^{1/2} < 1.\ n^{3/2}$
$f(n)= O(g(n))$

c) for all n, $f(n)=\Omega(g(n))$, c=1
for all n, $f(n) = O(g(n))$, c=2
$f(n) = \Theta(g(n))$

d) $\log n < \log 10n$
$f(n) < 1.\ g(n) \Rightarrow f(n)= O(g(n))$, c=1

$g(n)/f(n) < 20$
$g(n) < 20.\ f(n) \Rightarrow f(n)=\Omega(g(n))$, c=20
$f(n) = \Theta(g(n))$

e) $\log 2n < \log 3n$
$f(n) < 1.\ g(n) \Rightarrow f(n)= O(g(n))$, c=1

$g(n)/f(n) < 2$
$g(n) < 2.\ f(n) \Rightarrow f(n)=\Omega(g(n))$, c=2
$f(n) = \Theta(g(n))$

f) $10 \log n > 2 \log n$
$g(n) < 1.\ f(n) \Rightarrow f(n)=\Omega(g(n))$, c=1

$10 \log n < 6.\ 2 \log n$
$f(n) < 6.\ g(n) \Rightarrow f(n)= O(g(n))$, c=6
$f(n) = \Theta(g(n))$

g) $n^{0.01} > \log^2 n$
$n^{1.01} > n \log^2 n$
$g(n) < 1.\ f(n) \Rightarrow f(n)=\Omega(g(n))$, c=1

h) $f(n) / g(n) \geq 1$

$f(n) = \Omega(g(n))$, c=1

i) $\text{Lim}_{n\to\infty} f(n)/g(n) = \infty$

$f(n) = \Omega(g(n))$

j) $\text{Lim}_{n\to\infty} f(n)/g(n) = \infty$

$f(n) = \Omega(g(n))$

k) $\text{Lim}_{n\to\infty} f(n)/g(n) = \infty$

$f(n) = \Omega(g(n))$

l) $f(n) / g(n) = n^{1/2} / 5^{\log 2n} = \log_5 n^{1/2} / \log_2 n = (\log_2 n / 2\log_2 5) / \log_2 n$

$f(n) < 1.\ g(n) \Rightarrow f(n) = O(g(n))$, c=1

$g(n) < 6.\ f(n) \Rightarrow f(n) = \Omega(g(n))$, c=6

$f(n) = \Theta(g(n))$

m) for all $n>0$, $n2^n < 3^n$

$f(n) = O(g(n))$

n) $2^n < 2^{n+1}$

$f(n) < 1.\ g(n) \Rightarrow f(n) = O(g(n))$, c=1

$4.2^n = 2^{n+2} > 2^{n+1}$

$g(n) < 4.\ f(n) \Rightarrow f(n) = \Omega(g(n))$, c=4

$f(n) = \Theta(g(n))$

o) $g(n) = 2^n = 2.2.2\ldots2$

$f(n) = n! = 2.3.4\ldots n$

for all $n>1$, $g(n) < 2.\ f(n)$

$\Rightarrow f(n) = \Omega(g(n))$, c=2

p) $g(n) = 2^{(\log n)2} = n^{\log n}$

$n > \log n$

$f(n) < 1.\ g(n) \Rightarrow f(n) = O(g(n))$, c=1

q) $f(n) = \sum_{1 \text{ to } n} i^k = 1 + 2^k + \ldots + n^k < n.n^k$

$f(n)= O(g(n))$

use induction to prove this part

$(n-1)^{k+1} < c\sum_{1 \text{ to } n-1} i^k$

$(n-1)^{k+1} + cn^k > n^{k+1}$

$n^{k+1} < c\sum_{1 \text{ to } n} i^k$

$f(n)=\Omega(g(n))$

$f(n) = \Theta(g(n))$

2. a) if $c < 1$,

$1.1< g(n) <2.1$

$g(n) = \Theta(1)$

b) if $c=1$, $g(n) =n+1$

$1.n < g(n)<2.n$

$g(n) = \Theta(n)$

c) if $c> 1$,

$\lim_{n\to\infty} g(n)/ c^n = c/(c-1)$

$g(n) = \Theta(c^n)$

3. a) $F_n = F_{n-1} + F_{n-2} = 2^{0.5n-0.5} + 2^{0.5n-1} = 2^{0.5n} (2^{-0.5} + 2^{-1}) \geq 2^{0.5n} . 1\geq 2^{0.5n}$

b) $c = 0.8 / 0.9$

c) $F_n \geq 2^{cn}$

$F_{n+1} = F_n + F_{n-1} \geq 2^{cn} + 2^{cn-c} = 2^{cn} (1+2^{-c}) \geq 2^{cn+c}$

⇨ $c \leq \log ((1 +\text{root}(5)) / 2)$

4. $a_{k-1}n^{k-1} + \ldots + a_0 \leq cn^{k-1}$

$a_k n^k + a_{k-1}n^{k-1} + \ldots + a_0 \leq cn^{k-1} + a_k n^k = c1\ n^k$

5. The three stacks are marked S1, S2, and S3.

S1 is the original unsorted stack, S2 will eventually have the result, and S3 is the working stack. To understand the below, imagine/assume that

(a) S2 already has some of the numbers stored in the sorted order; initially S2 is empty
(b) S1 has the remaining unsorted numbers.
(c) S3 is empty.

We do insertion sort, by popping a number X from top of S1 and pushing it to S3. Then, we insert this number X in an appropriate place in S2 by:

- Pushing a "marker" into S1.
- Popping numbers from S2 into S1, until the top of S2 is just less than X (the top of S3).
- At this point, we push X from S3 to S2.
- Then, we pop numbers from S1 to S2, until the "marker" on S1.

6. Assume n is a power of 2. Now, construct a binary tree over the given numbers, where the given numbers are the leaves of the tree. At each interior tree-node i, we store the sum of the leaves that are descendants of i.

a) Add(i,y) can be done in O(log(n)) time, because adding a value to a leaf node only affects the sum stored at its ancestors, which are O(log n) in number.
b) Partial-Sum(i) can also be done in O(log n) time, by adding up the values of the following interior tree-nodes: Follow the tree path to the i^{th} leaf. For each node I on the traversed path, add the value stored at the left-child of i.