Problem 1.1

a)

$n = 500$

$N = 1000$

$N = 2n$

$$\frac{T(2n)}{T(n)} = \frac{C_M \left(\frac{1}{3}\right)(2n)^3}{C_M \left(\frac{1}{3}\right) n^3}$$

$$= \frac{8n^3}{n^3} = \boxed{8}$$

b)

The running time can be estimated by solving system of order n on old computer and order N on the new computer.

$$T_{old}(n) = \frac{1}{3}n^3$$

$$T_{new}(N) = 10^{-3} \frac{1}{3}N^3$$

$$\frac{1}{3}n^3 = 10^{-3}\frac{1}{3}N^3$$

$$\boxed{\frac{N}{n} = 10.}$$

Problem 1.2

$$5\log_2(n+100)^{10} < \log_2^2 n < \sqrt[3]{n} < 0.001n^4 + 3n^3 + 1$$
$$< 3^n < 2^{2n} < (n-1)!$$

Problem 1.3

a) $\displaystyle\sum_{K=1}^{1000} K - 2\sum_{K=1}^{500} K = \frac{(1000 \times 1001)}{2} - (500 \times 501) = \boxed{250\,000}$

b) $\displaystyle\sum_{K=1}^{10} 2^k = \frac{2^1 - 2^{11}}{1-2} = \frac{2-2048}{-1} = \boxed{2046}$

c) $\boxed{n-1}$

d) $\displaystyle\sum_{i=3}^{n+1} i = \sum_{i=1}^{n+1} i - (1+2) = \boxed{\frac{(n+1)(n+2)}{2} - 3}$

e) $\displaystyle\sum_{i=1}^{n} \sum_{j=1}^{n} ij = \sum_{i=1}^{n} i \sum_{j=1}^{n} j = \frac{n(n+1)}{2} \times \frac{n(n+1)}{2} = \boxed{\frac{n^2(n+1)^2}{4}}$

Problem 1.4

a) This algorithm computes the difference between maximum value and minimum value in an array.

b) Basic operation is the comparison between current value and minimum and maximum value.

c) Each comparison is performed $n-1$ times

d) $2(n-1)$

e) $O(n)$

f) No better algorithm exists since all elements of the array must be exaimed to find the max & min value.

a) This algorithm return true if array is symmetric, otherwise it return false.

b) The basic operation is the comparison of $A[i,j]$ and $A[j,i]$

c) $(n-1) + (n-2) + \dots 1$

d) $\frac{(n-1)n}{2}$

e) $O(n^2)$

f) An array is symmetric when the elements above the diagonal is equal to the diagnol below. All of the elements beside the diagnol need to be exaimed. There are $\frac{n(n-1)}{2}$ elements above and below the diagonal, therefore, $\frac{n(n-1)}{2}$ is the minmum number of comparison needed to be execute.

Problem 1.6

$$L(n) = \begin{cases} 1, & n=1 \\ 2, & n=2 \\ L(n-1)+L(n-2), & n>2 \end{cases}$$

Problem 1.7

```
procedure Swap (string)
  if length(string) = 0 || length(string) = 1
      return string.
  else
      a ← first character of string
      b ← second character of string
      string ← string without a & b
      return "a + b" + string
```