Homework 1

1. Step1 assign n to positive integer. left = 0. right = n/2, answer

   Step2 set up while loop left less and equal right
   Initialize middle = (lett + right) / 2

   Step3 if middle * middle = n
   answer = middle
   return answer

   Step4 else if middle * middle < n
   left = middle + 1
   answer = left

   Step5 else
   right = middle - 1
   answer = right

   step 6: repeat nutil left greater than right

2. Step1 initialize pointer i for m list and j for n list
   assign m and n data

   Step2 set up while loops i < m's number of data,
   j < n's number of data

   step3 if m[i] < n[j]
   i = i+1

   step4 else if m[i] > n[j]
   j = j+1

   step5 esle m[i] = n[j]
   append m[i] into new list

   Step 6 repeat nutil i > m's size or j > n's size

the maximum number of comparison is m+n ※

step 1
3 set up a function to find binary representation of a positive decimal integer. integer type data is assignment.

step 2. set up an empty vector for new decimal integer

step 3, set up while loop data > 0

step 4. initialize remainder of data, and then push-back to new vector

data = data / 2 : find the quotient

step 5 repeat until data = 0. the new vector is decimal integer

4. step 1 setup a function to find the closet distance between two elements. first initialize minimum distance and assign it infinite.

step 2 for loop from i = 0 to array size -1 and another for loop from j = 0 to array size -1.

step 3 if i is not j and distance is less than min distance.

step 4 return mindistance until the end of nested loop

5. step 1 set up a function to find anagram between two strings

step 2 initialize n1 is the lenguth of string one.
initialize n2 is the lenguth of string two.

step 3. n1 is not n2 , return false

step 4 sort the string one and string two by alphabet.

step 5 for loop int i=0, to string one size.
if string1[i] is not equal to string2[i] return fals
else retun True

6. ① The basic operation of adding two m x n matrices is adding corresponding elements from both matrices and place new elements in a new matrix
ex A[i][j] , B[i][j] i is row index, j is column index

$A[i][j] + B[i][j] = C[i][j]$

② In order to read A matrix for (i=0 to n-1)
for (j=0 to n-1)

It takes n^2 times for one matrix

③ But it takes $2n^2$ times for two matrices in total

7. ① The basic operation of mulitiplying two $n \times n$ matrices, is time corresponding elements from both matrices and place new elements in ~~the~~ a new matri

A[i][j] × B[i][j] = C[i][j]

② In order to read A matrix, use nested loops
for (i=0, to n-1)
~~for(j=0 to n-1)~~

It takes n^2 times for one matrix

③ It takes $2n^2$ times for two matrices in total

8. Step1 set up linear search function, assign arr, and key data

step2 for loop iterate array. for(int i=0, i< array size)

Step3 repeat nutil if array[i] = key return true
else return false,

Algorithm linear sort (arr[], int data)
for (i=0, to size -1)
if(data == arr[i])
return true.

$$\sum_{i=0}^{n-1} 1 = n-1+1 = n \qquad O(n)$$

9. Let's take bubble sort as example. There is a need to compare the adjacent elements of its input. It slows down the algorithm, so for this reason it won't be a worthwhile addition

10. a. $n(n+1) = n^2+1 \approx n^2$ has the same order of growth as $2000n^2$

b. n^2 has lower order of growth than n^3

c. $\log n$ has the same order of growth as $\ln n$

d. 2^n has higher order of growth than $(n-1)!$

11. a $\frac{(1+999)\times 500}{2} = 250{,}000$ #

b. $\sum_{i=3}^{n+1} i = \sum_{i=0}^{n+1} i - \sum_{i=0}^{2} i = \frac{(n+1)(n+2)}{2} - 3 = \frac{n^2+3n-4}{2}$ #

c. $\sum_{i=0}^{n-1} i(i+1) = \sum_{i=0}^{n-1} i^2+i = \sum_{i=0}^{n-1} i^2 + \sum_{i=0}^{n-1} i = \frac{(n-1)(n)(2n-1)}{6} + \frac{(n-1)n}{2}$

$= \frac{(n^2-1)n}{3}$ #

d. $\sum_{i=1}^{n}\sum_{j=1}^{n} ij = \sum_{i=1}^{n} i \sum_{j=1}^{n} j = \sum_{i=1}^{n} i \times \frac{(1+n)n}{2} = \left(\frac{n(n+1)}{2}\right)^2$ #

12. a. $\sum_{i=0}^{n-1}(i^4+2i+1) = \sum_{i=0}^{n-1} i^4 + 2\sum_{i=0}^{n-1} i + \sum_{i=0}^{n-1} 1$

$\in \Theta(n^6) + \Theta(n^3) + \Theta(n) = \Theta(n^6)$

b. $\sum_{i=2}^{n-1} \log_2 i^2 = \sum_{i=2}^{n-1} 2\log_2 i = 2\sum_{i=2}^{n-1} \log_2 i = 2\sum_{i=1}^{n} \log_2 i - 2\log_2 n$

$\in 2\Theta(n\log n) - \Theta(\log n) = \Theta(n\log n)$

c. $\sum_{i=1}^{n} (i+1)2^{i-1} = \sum_{i=1}^{n} i2^{i-1} + \sum_{i=1}^{n} 2^{i-1} = \frac{1}{2}\sum_{i=1}^{n} i2^i + \sum_{j=0}^{n-1} 2^j$

$\in \Theta(n2^n) + \Theta(2^n) = \Theta(n2^n)$

d. $\sum_{i=0}^{n-1}\sum_{j=0}^{i-1}(i+j) = \sum_{i=0}^{n-1}\left(\sum_{j=0}^{i-1} i + \sum_{j=0}^{i-1} j\right) = \sum_{i=0}^{n-1}\left(i^2 + \frac{(i-1)i}{2}\right)$

$= \sum_{i=0}^{n-1}\left[\frac{3}{2}i^2 - \frac{1}{2}i\right] = \frac{3}{2}\sum_{i=0}^{n-1} i^2 - \frac{1}{2}\sum_{i=0}^{n-1} i = \Theta(n^3) - \Theta(n^2)$

$= \Theta(n^3)$

13. a summation of n^2, from 1 to n.

$$\sum_{i=1}^{n} i^2$$

b. $S = S + i^2$ multiplication

c. n times

d. $C(n) \in O(n)$

14 use the formula $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$ to compute ~~the sum in~~ $\Theta(1)$ time

15 a. $x(n) = 3x(n-1) \quad n > 1$

$= 3[3x(n-2)] = 3^2 x(n-2)$

$= 3^2[3x(n-3)] = 3^3 x(n-3)$

$= 3^3[3x(n-4)] = 3^4 x(n-4)$

$\vdots$

$= 3^i x(n-i)$

$= 3^{n-1} x(1) = 3^{n-1} \cdot 4$

b. $x(n) = x(n-1) + n \quad n>0 \quad x(0)=0$

$= [x(n-2) + (n-1)] + n = x(n-2) + (n-1) + n$

$= [x(n-3) + (n-2)] + (n-1) + n = x(n-3) + (n-2) + (n-1) + n$

$\vdots$

$= [x(n-i) + (n-i+1)] + (n-i+2) \ldots n$

$= x(0) + 1 + 2 + 3 \ldots n = x(0) + \frac{(1+n) \times n}{2} = \frac{(1+n) \times n}{2}$

c solve $x(2^k) = x(2^k/2) + 2^k = x(2^{k-1}) + 2^k$

$= x(2^{k-2}) + 2^{k-1} + 2^k$

$= x(2^{k-3}) + 2^{k-2} + 2^{k-1} + 2^k$

$= x(2^{k-i}) + 2^{k-i+1} + 2^{k-i+2} \ldots 2^k$

$= x(2^{k-k}) + 2^1 + 2^2 + 2^3 \ldots 2^k$

$= 1 + 2^1 + 2^2 \ldots 2^k$

$= 2^{k+1} - 1 = 2^k \cdot 2 - 1 = 2n - 1$

d. solve $x(3^k) = x(3^{k-1}) + 1$

$= x(3^{k-2}) + 1 + 1 = x(3^{k-2}) + 2$

$= x(3^{k-3}) + 3$

$= x(3^{k-i}) + i$

$= x(3^{k-k}) + k$

$= x(1) + k$

$= 1 + \log_3 n$

16 a. $Q(n) = Q(n-1) + (2 * n-1 \quad Q(1)=1 \quad n>1$

$= Q(n-1) + 2n - 1$

$Q(2) = Q(1) + 2\cdot2-1 = 4$

$Q(3) = Q(2) + 2\cdot3-1 = 9$

$Q(4) = Q(3) + 2\cdot4-1 = 16$

it looks like $Q(n) = n^2$

If $Q(n) = Q(n-1) + 2n-1$

$(n-1)^2 + 2n - 1 = n^2 + \cancel{1} - \cancel{2n} + \cancel{2n} - \cancel{1}$

$= n^2$

b. $M(n) = M(n-1) + 1 \qquad M(1) = 0 \quad n>1$

$M(2) = M(1) + 1 = 1$

$M(3) = M(2) + 1 = 2$

$M(4) = M(3) + 1 = 3$

$M(n) = n-1$

c. $C(n) = C(n-1) + 3 \quad n>1 \quad C(1)=0$

$C(2) = C(1) + 3 = 3$

$C(3) = C(2) + 3 = 6$

$C(4) = C(3) + 3 = 9$

$C(n) = 3(n-1)$