ADA homework 2
b03902089 林良翰

problem 1.(1)

```
function binomialNumber(n, m)
        binoNum[n+1][n+1] = {0}
        binoNum[0][0] = 1
        for i from 1 to n :
                for j from 0 to i :
                        binoNum[i][j] = binoNum[i-1][j] + binoNum[i-1][j+1]
                end for
        end for
        return binoNum[n][m]
```

problem 1.(2)

```
function catalanNumber(n)
        count[n+1][n+1] = {0}
        count[0][0] = 0
        for i from 1 to n :
                for j from 1 to n :
                        count[i][j] = count[i-1][j] + count[i][j-1]
                end for
        end for
        return count[n][n]
```

problem 1.(3)

by the definition of "H(n, g)", we can build the table

groups / number	1	2	3	4	5	6	7
1	1						
2	1	1					
3	1	1	1				
4	1	2	1	1			
5	1	2	2	1	1		
6	1	3	3	2	1	1	
7	1	3	4	3	2	1	1

==========

```
function partionNumber(n)
        partNum[n+1][n+1]
        partNum[1][1] = partNum[2][1] = partNum[2][2] = 1
        for num = 3 to n :
                for group = 1 to num :
                        partNum[num][group] = sum of (partNum[num-group][1] to partNum[num-
group][group])
```

```
		end for
	end for
	return sum of (partNum[n][1] to partNum[n][n])
```

problem 2.(1)
see problem 2.(2)

problem 2.(2)
set f(n) to be the number of hidden code of HH-code :
f(n) =

(a). 1, if length = 0
(b). 2, if length = 1
(c). $f(n-1) \times 2 - f$(the previous place same element going to add)

========== code :

```
function findHidden(code)
	hiddenNum[length+1]
	for pos = 1 to max_length :
		if pos == 1 :
			hiddenNum[pos] = 1
		else
			for pre = pos-1 to 1 :
				if code[pre] == code[pos] :
					break
				end if
			end for
			hiddenNum[pos] = hiddenNum[pos−1]*2 − hiddenNum[pre-1]
		end if
	end for
	return hiddenNum[length]-1
```

problem 2.(3)

K N		0	1	2	3	4
0	null	0	0	0	0	0
1	1	0	1	0	0	0
2	0	0	2	1	0	0
3	1	0	2	3	1	0
4	1	0	2	3	3	1

problem 3.(1).1
Each multiplication of element need **O**(m) time :
ex: [a b; c d]*[a b; c d]
(a*a + b*c) ⇒ m = 2, need to add m times

And we have m^2 elements ⇒ **O**(m^3)

Last, we compute A^n by dividing it into half again and again ex: $A^n = A^{n/2} \times A^{n/2}$

$= (A^{n/4} \times A^{n/4}) \times A^{n/2}$

$= \dots$

So, we need ⇒ **O**($m^3 \log_2 n$) time to calculate A^n

problem 3.(1).2