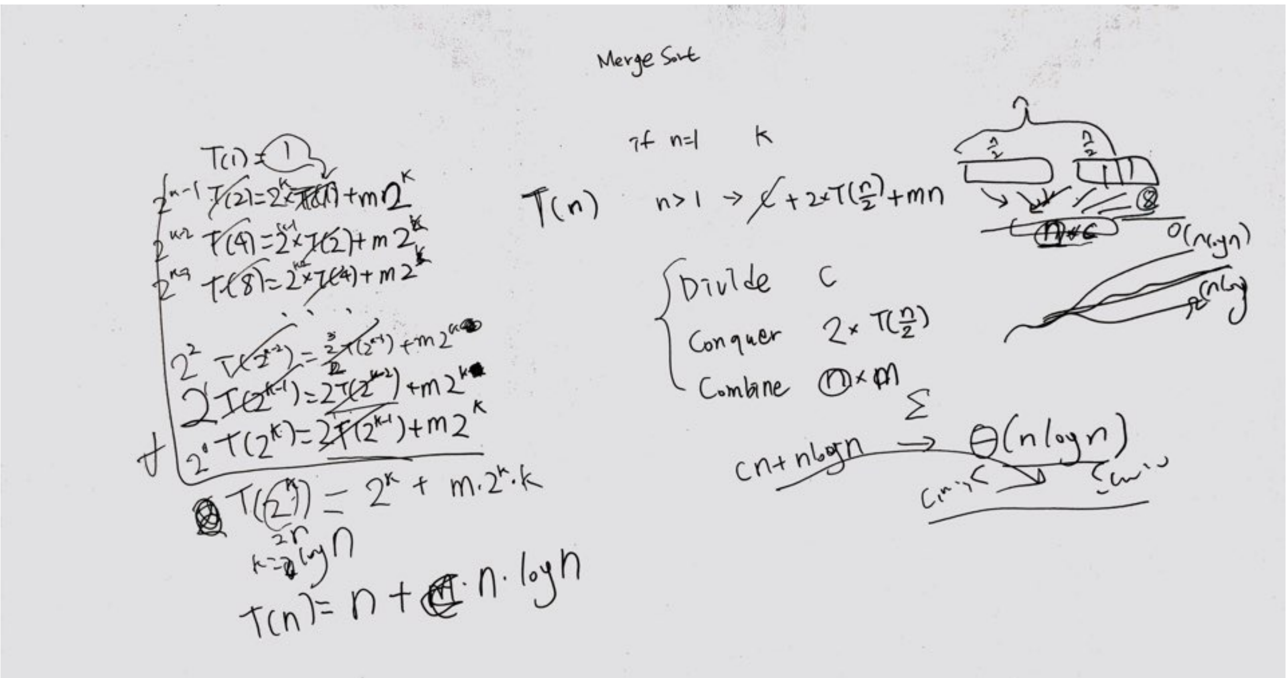
[희수]

Merge Sort Time Complexity explain.



Merge Sort

if n=1 k

$T(n)$ n>1 → $\cancel{c} + 2\times T(\frac{n}{2}) + mn$

Divide c

Conquer $2\times T(\frac{n}{2})$

Combine $n \times m$

Σ

$cn + n\log n \rightarrow \Theta(n\log n)$

$T(1) = 1$

2^{k-1} $T(2) = 2^k \times T(1) + m2^k$

2^{k-2} $T(4) = 2^{k-1} \times T(2) + m2^k$

2^{k-3} $T(8) = 2^{k-2} \times T(4) + m2^k$

$\vdots$

2^2 $T(2^{k-2}) = 2^3 T(2^{k-3}) + m2^k$

2^1 $T(2^{k-1}) = 2T(2^{k-2}) + m2^k$

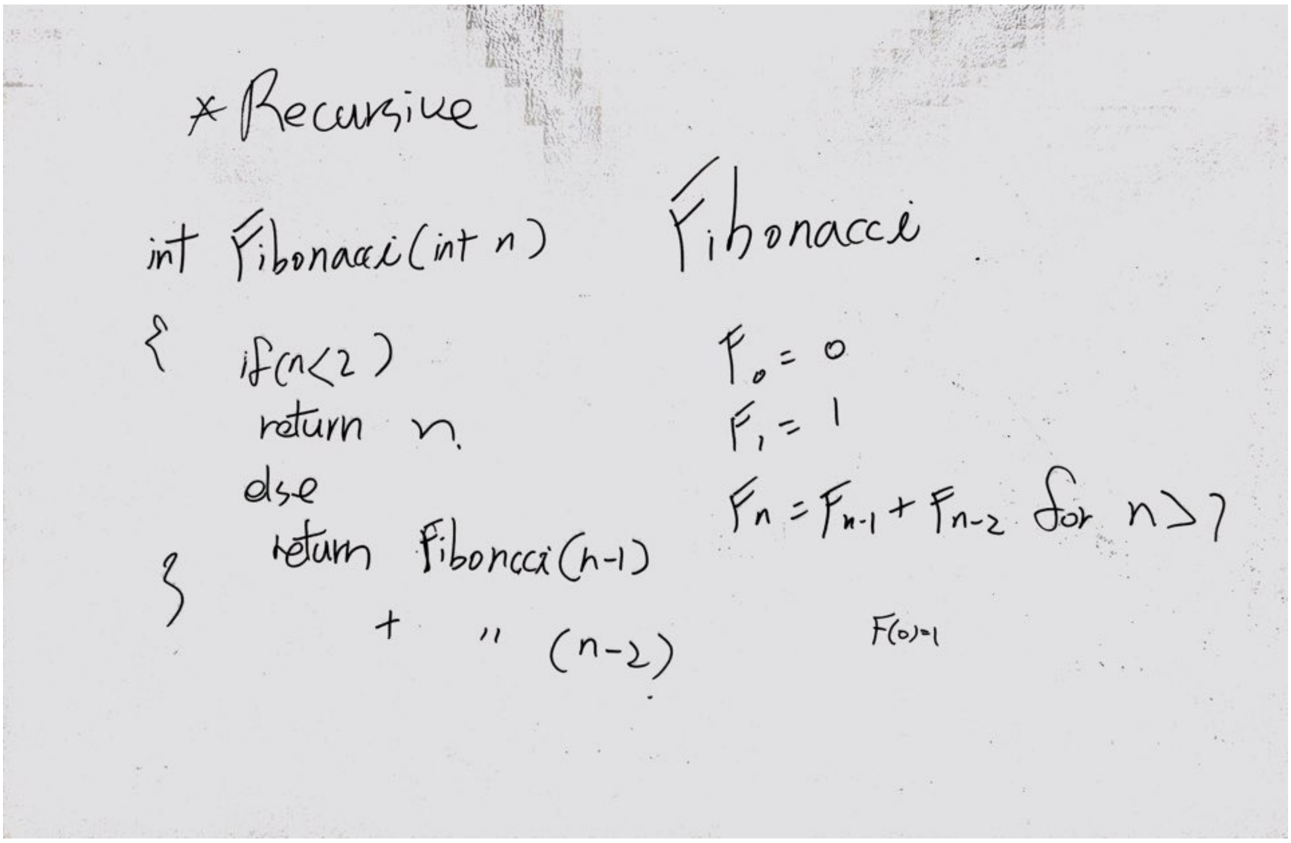
2^0 $T(2^k) = 2T(2^{k-1}) + m2^k$

$T(2^k) = 2^k + m \cdot 2^k \cdot k$

$2^k = n$, $k = \log n$

$T(n) = n + m \cdot n \cdot \log n$

[동준]
Fibonacci Explain but It was’t enough
So I’m preparing again



* Recursive

```
int Fibonacci(int n)
{  if(n<2)
     return n.
   else
     return Fiboncci(n-1)
          +    "    (n-2)
}
```

Fibonacci .

$F_0 = 0$

$F_1 = 1$

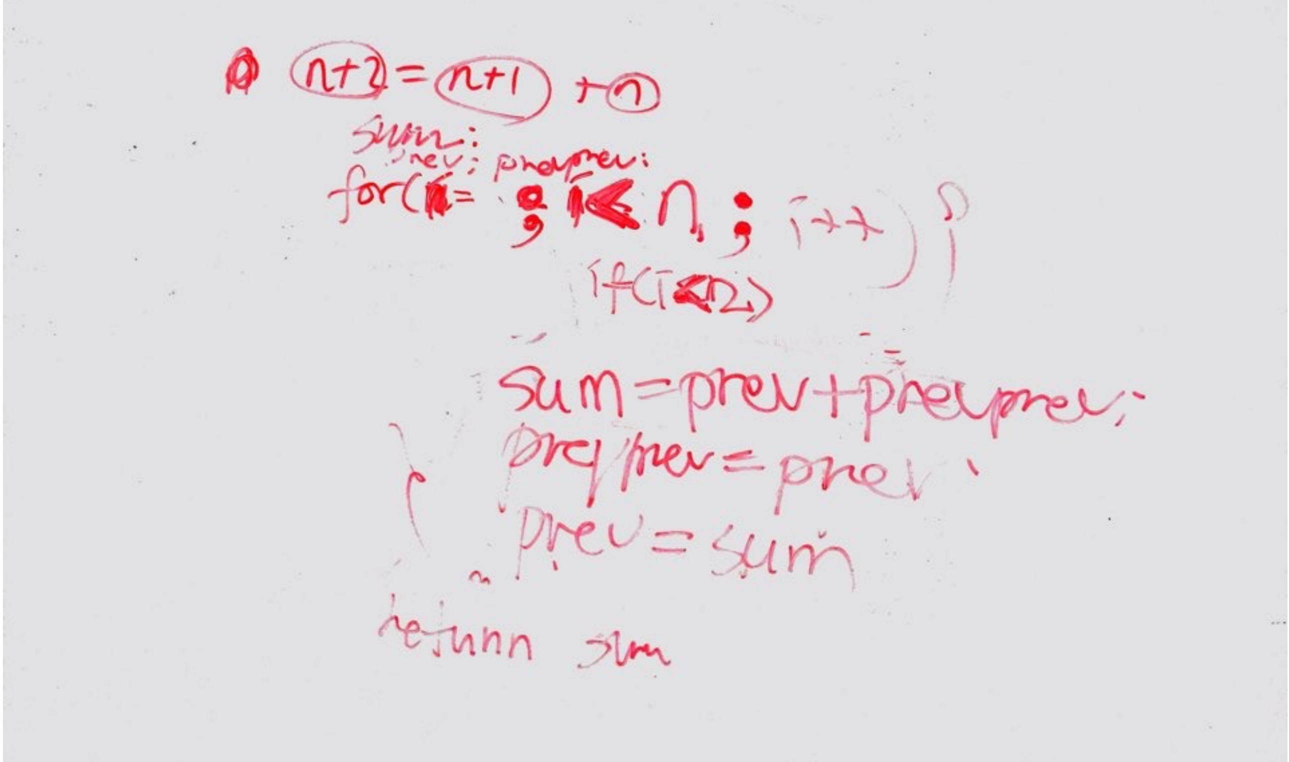
$F_n = F_{n-1} + F_{n-2}$ for $n > 7$

$F(0)=1$

<AS by DongJun>

```
int Fibonacci(int n){

    int result;
    int i;
    int * FiboArr=(int*)malloc(sizeof(int)*n);
    //malloc된 값을 해제할 free는 main에서 해제합니다!

     FiboArr[0]=0;
     FiboArr[1]=1;

    for(i=2; i<=n; i++)
     {
         result=FiboArr[i-1]+FiboArr[i-2];
         FiboArr[i]=result;
     }
    return FiboArr[i];

}
```

[기범]

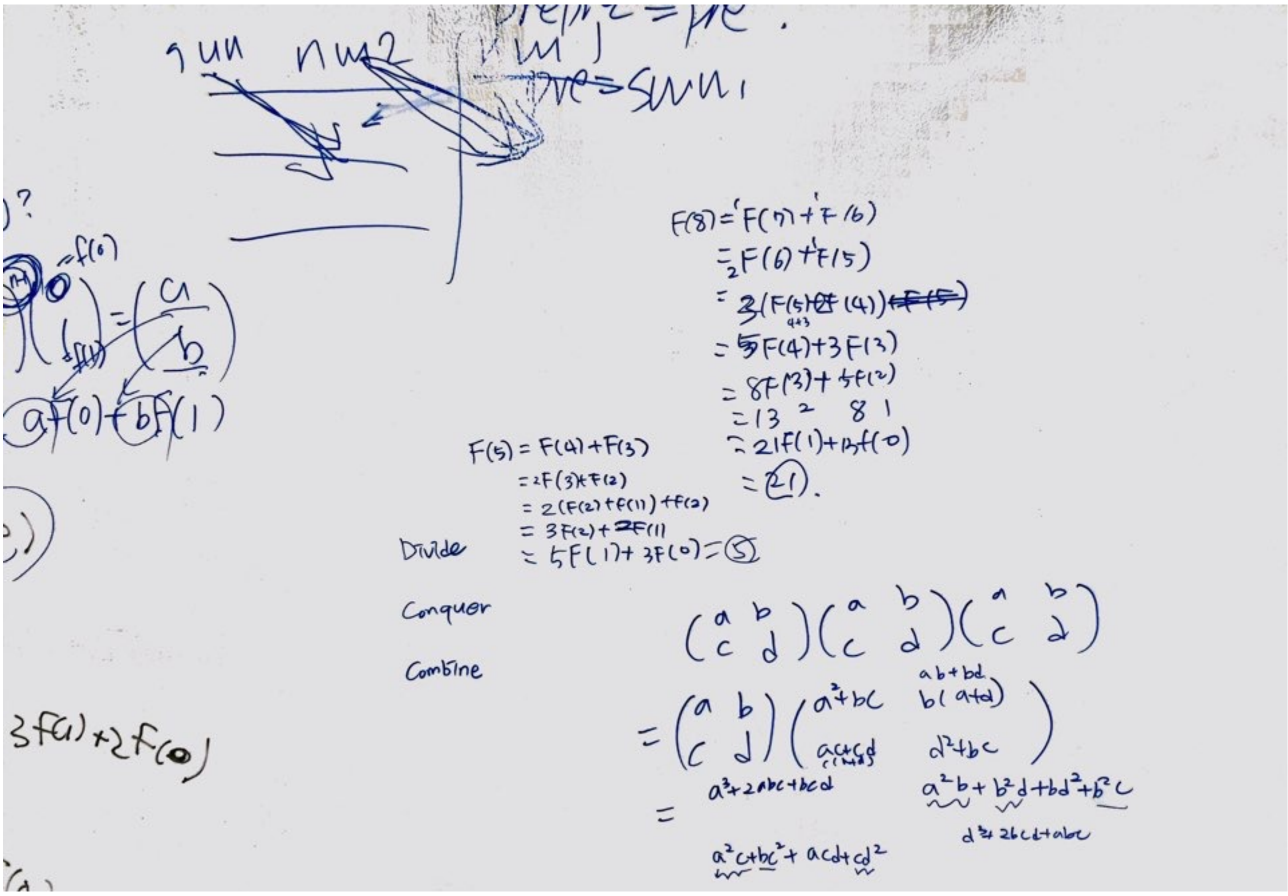


```
(n+2) = (n+1) + (n)
sum;
prev; preprev;
for(i= 2; i<=n; i++) {
    if(i<=2)

    sum=prev+preprev;
    preprev=prev
    prev=sum
}
return sum
```

<AS by DongJun>

```
      int Fibonacci(int n)
     {

             if (n == 0)
                   return 0;
             if (n == 1)
                   return 1;

        int result=0;
        int prevPrev=0;
        int prev=1;

        for(int i=2; i<=n; i++)
        {
            result = prev + prevPrev;
            prevPrev = prev;
            prev = result;
         }

        return result;
   }
```

[희수]
Divide & Conquer



[규원]
Divide & Conquer