

Group Activity 12

(3인 혹은 4인으로 팀을 구성하여 아래의 문제를 푼다. 팀 구성은 매 시간마다 달라져도 된다.)

팀원1: _____

팀원2: _____

팀원3: _____

팀원4: _____

1. 다음의 각각의 함수의 최악의 경우의 시간복잡도를 점근적(asymptotic) 표기법으로 나타내면? 이유는?

```
int fun1(int n) {  
    int count = 0;  
    for (int i = 0; i < n; i++)  
        for (int j = i; j > 0; j--)  
            count = count + 1;  
    return count;  
}
```

```
double fun2( int n ) {  
    double sum = 0;  
    for (double i = 1.0; i < n; i *= 1.5)  
        sum += i;  
    return sum;  
}
```

```
int fun3( int n, int data[] ) {  
    int sum = 0;  
    for (int i = 1; i < n; i *= 3)  
        sum += data[i];  
    return sum;  
}
```

```

/* 단 max와 배열 data에 저장된 값은 모두 양수 */
void fun4( int n, int max, int data[] ) {
    int j = 0, sum = 0;
    for (int i=0; i<n; i++) {
        sum += data[i];
        while(sum >= max)
            sum -= data[j++];
    }
}

```

```

/* 배열 data에 n개의 정수들이 오름차순으로 정렬되어 있음 */
int fun5( int n, int K, int data[] ) {
    int count = 0;
    for (int i=0; i<n; i++) {
        int result = binary_search(n, data, K-data[i]); /* 이진검색을 수행한다. */
        if (result != -1)
            count++;
    }
    return count;
}

```

```

/* 배열 A와 B에 각각 m개와 n개의 정수가 오름차순으로 정렬되어 저장. 그 외에 어떤 가정도 없음 */
void fun6(int m, int A[], int n, int B[]) {
    for (int i=0; i<m; i++) {
        for (int j=1; j<n; j*=2) {
            if (A[i] <= B[j])
                break;
        }
    }
}

```

```
int fun7(int n). {
    int count = 0;
    for (int i = n; i > 0; i /= 2)
        for (int j = 0; j < i; j++)
            count += 1;
    return count;
}
```

```
void fun8(int n, int arr[]). {
    int i = 0, j = 0;
    for(; i < n; ++i)
        while(j < n && arr[i] < arr[j])
            j++;
}
```

2. 프로그램에서 리스트(list)를 표현하는 대표적인 2가지 방법은 배열과 연결리스트이다. 또한 각각의 경우 데이터를 크기순으로 정렬해서 저장할 수도 있고 그렇지 않을 수도 있으므로 총 4가지 방법을 생각해 볼수 있다. 길이가 N 인 리스트를 이러한 4가지 방법으로 저장했을 때 다음 각각의 연산의 최악의 경우 시간복잡도는? 이유는?

- 검색: 어떤 값이 리스트에 포함되어 있는지 검사한다.
- 삽입: 새로운 하나의 값을 리스트에 추가한다.
- 삭제: 어떤 값을 리스트로 부터 삭제한다 (단 리스트에서 삭제할 값의 위치를 찾는데 걸린 시간은 제외한다.)

자료구조		검색(search)	삽입(insert)	삭제(remove)
배열	정렬 안함			
	오름차순으로 정렬함			
연결리스트	정렬 안함			
	오름차순으로 정렬함			