Swap 알고리즘

1. 임시공간할당방식 (temp변수 사용)

일반적인 방법으로 임시 저장 변수 temp 사용.

Temp = a;

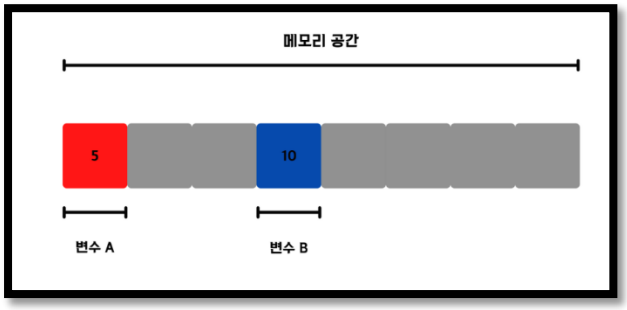
a = b;

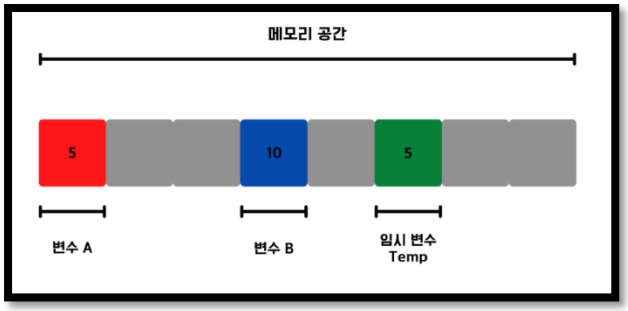
b = temp;

=> 저장 공간이 하나 더 필요하다. But, 단순 대입연산이기 때문에 타 연산에 비해 속도가 빠르다.

함수의 경우 call by reference를 이용

주소를 매개변수로 넘겨주어 함수 구현부에서 주소를 찾아가 값을 바꿈. 매개변수와는 상관없이 저장된 값이 바뀌게 됨.





2. 사칙연산

1) +,- 활용

a = a + b;

b = a – b;

a = a – b;

2) *, / 활용

a = a * b;

b = a / b;

a = a / b;

3. XOR(배타적 논리합) 연산

a = a ^ b;

b = a ^ b;

a = a ^ b;

연산 A, B 는 서로 반대되는 성질을 가져야함 (ex) (-, +) , (*, /)

사칙연산의 경우 정수, 실수에서 가능하지만, 비트 연산은 정수에서만 가능하다.

x = 1, y = 2 일 때.

a) x = x ^ y = 3		0 0 0 1
	XOR	0 0 1 0
	=	0 0 1 1
b) y = x ^ y = 1		0 0 1 1
	XOR	0 0 1 0
	=	0 0 0 1
c) x = x ^ y = 2		0 0 1 1
	XOR	0 0 0 1
	=	0 0 1 0

결과 : x = 2, y = 1

swap() : 매개변수

-swap은 algorithm 라이브러리의 있는 함수로서 두 변수의 저장되어 있는 값을 서로 교환한다.

```
#include <iostream>
#include <algorithm>
using namespace std;

int main() {

    int a = 3, b = 5;
    cout << "a: " << a << ", b: " << b << endl;
    swap(a, b);
    cout << "a: " << a << ", b: " << b << endl;

    return 0;
}
```

```
a: 3, b: 5
a: 5, b: 3
```

swap_ranges() 함수

-swap_ranges 함수는 반복자를 인자로 받아, 지정한 범위의 값들을 서로 교환한다.

-인자로 (시작 반복자, 종료 반복자, 교환 시작 반복자) 로서

시작 반복자 ~ 종료 반복자 만큼의 크기만큼 교환 시작 반복자에서 시작하여 교환을 한다.

```
#include <iostream>
#include <algorithm>
using namespace std;

int main() {

    int a[5] = { 1, 2, 3, 4, 5 };
    int b[5] = { 6, 7, 8, 9, 10 };
    swap_ranges(a, a + 3, b);

    cout << "a:";
    for (int i = 0; i < 5; i++) cout << ' ' << a[i];
    cout << endl << "b:";
    for (int i = 0; i < 5; i++) cout << ' ' << b[i];
    cout << endl;

    return 0;
}
```

```
a: 6 7 8 4 5
b: 1 2 3 9 10
```