**STL 기초 – search**

1. FwdIt1 search(FwdIt1 first1, FwdIt1 last1, FwdIt2 first2, FwdIt2 last2 [, BinPred F]);
2. FwdIt1 find\_end(FwdIt1 first1, FwdIt1 last1, FwdIt2 first2, FwdIt2 last2 [,BinPred F]);
3. FwdIt1 search\_n (FwdIt1 first1, FwdIt1 last1, Size count, const Type& val[, BinPred F]);

🡺 반복자 구간에서 다른 구간 전체가 발견되는 지점을 검색한다.

\*Search 예제

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main()

{

int ar1[]={3,1,4,1,5,9,2,6,5,3,5,8,9,9,9,3,2,3,1,5,9,2,6,4,3};

int ar2[]={1,5,9};

int \*p;

p=search(&ar1[0],&ar1[25],&ar2[0],&ar2[3]);

if (p!=&ar1[25]) {

printf("%d번째에서 구간이 발견되었습니다.\n",p-ar1);

}

p=find\_end(&ar1[0],&ar1[25],&ar2[0],&ar2[3]);

if (p!=&ar1[25]) {

printf("%d번째에서 구간이 발견되었습니다.\n",p-ar1);

}

p=search\_n(&ar1[0],&ar1[25],3,9);

if (p!=&ar1[25]) {

printf("%d번째에서 3연속의 9를 발견했습니다.\n",p-ar1);

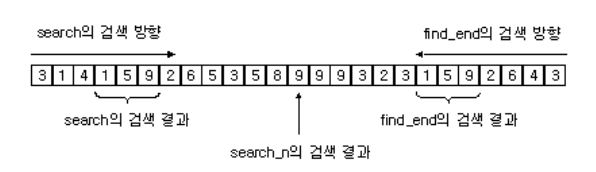
}

}

🡺search는 전체 구간의 앞쪽에서부터 검색

🡺find\_end는 전체 구간의 뒤쪽에서부터 검색을 한다. (search,find\_end 두 함수가 검색 시작 방향만 틀릴 뿐이다)

🡺search\_n은 반복자 구간에서 val 값이 count번 연속으로 나타나는 지점을 찾는다



순방향으로 검색한다면 search 함수를 사용하고 역방향으로 검색한다면 find\_end 함수를 사용하면 된다