**Chapter\_13\_탐색**

**- 색인 순차 탐색에서 왜 이진 탐색이 아닌 순차 탐색을 사용할끼?**

색인 순차 탐색은 큰 데이터 집합을 디스크 같은 보조기억장치에 저장했을 때 효율적으로 탐색하려 고안된 기법이다. 즉, 비교 횟수보다 디스크 접근 비용이 훨씬 크다고 보는 모델이다.

블록 단위 접근에서 진짜 큰 비용은 블록을 메모리로 불러오는 과정이지, 블록 내부 탐색 연산이 아니다. 블록 크기가 작으면 순차 탐색이 단순하고 빠르며, 캐시 친화적이다.

**- 캐시 친화적이라는 의미는?**

CPU는 메모리(RAM)보다 훨씬 빠르다. 그래서 CPU 옆에 작고 빠른 메모리(캐시)를 두고 자주 쓰거나 앞으로 쓸 가능성이 높은 데이터를 미리 가져다 둔다. 메모리에서 데이터를 가져올 때는 보통 한 덩어리씩 읽어온다.

순차 탐색은 데이터를 앞에서부터 차례대로 접근하므로, 메모리에서 블록을 읽을 때 그 다음 필요한 데이터가 이미 캐시에 올라와 있을 확률이 매우 높다.

그러나 이진 탐색은 차례대로 접근하지 않으므로 접근하는 위치가 메모리 상에서 멀리 떨어져 있을 수 있다. 따라서 CPU가 매번 새로운 캐시 라인을 불러와야 한다.

즉 캐시 친화적이란 CPU캐시에 잘 맞게 동작해서 메모리 접근이 빠르게 일어난다는 뜻이다.

**- C언어 매크로 함수 주의점**

#define MAX(a,b) (a)>(b) ? (a) : (b)

이렇게 매크로 함수를 선언했다고 하자. 만약 다른 함수에서

1 + MAX(a,b) 혹은 MAX(a,b) + 1 을 사용할 경우 예상하지 못한 값이 나오게 된다.

그 이유는 컴파일러가 저 코드를 읽을 때 1+a > b ? 와 a > b ? a : b+1로 읽어 원래의 의도와 다르게 계산되기 때문이다. 따라서 매크로 함수를 사용할 때는

#define MAX(a,b) ((a)>(b) ? (a) : (b))

와 같이 괄호로 한번 더 묶어주어야 한다.