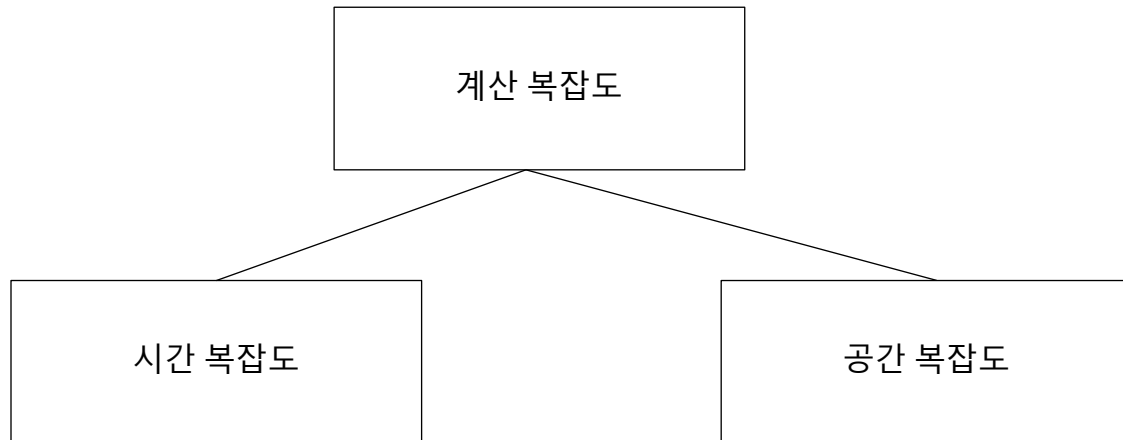


시간복잡도

2022/01/28 김기태



시간 복잡도 : 어떤 알고리즘의 소요 시간을 정확히 평가할 수 없으므로 이 알고리즘이 연산을 수행하는 횟수를 대략적으로 나타냄. 이를 $O(n)$ 으로 표기하는데 이를 빅 오 표기법이라고 한다. 이를 구하는 법은 예를 들어 입력의 개수가 n 인 반복문을 통해 $3n+1$ 번의 연산을 수행한다고 하면 대체적으로 선형적으로 증가하기때문에 $O(n)$ 이라고 표시한다.

공간 복잡도 : 어떤 알고리즘이 수행되는 과정에서 필요한 메모리 사용량을 나타냄. 시대가 흐름에 따라 보통은 중요하게 고려하지 않지만 *동적 계획법, 임베디드 시스템에서는 공간복잡도를 고려하여 알고리즘을 설계함.

*동적 계획법 : 피보나치 수열처럼 그 이전의 답을 메모리에 올려놓고 다음 답을 구하는 방식으로 만약 피보나치 수열을 계산할 때, $\text{fibo}(10)$ 을 구한다고 하면 이를 재귀 함수를 이용하여 답을 구하는 과정에서 $\text{fibo}(1)$ 이 구해졌다면 메모리에 저장해 두고 다음에 $\text{fibo}(1)$ 이 사용될 때 다시 가져다 쓰는 방식. 저장 해두고 다시 가져다 쓰는 방식을 '메모이제이션(memoization)' 이라고 한다.

코딩테스트에서의 Tip

- 보통의 최신 컴퓨터는 1 초에 '대략' 10 억(1,000,000,000)회의 연산을 한다고 한다. 만약 코딩테스트에서 3 초의 제한시간을 주었고, 데이터(n) 개수의 범위 $0 < n < 100,000$ (10 만)가 주어졌다고 해보자. 그렇다면 이중 for 문을 사용할 경우에 10,000,000,000(100 억)번의 연산이 수행이 되고 이는 10 초에 해당하므로 제한시간을 초과하게 될 수 있다.
- 그리고 이를 어떻게 최적화를 하여도 10 만개의 데이터를 이중 for 문을 사용했다면 '최소' 100 억번의 연산이 수행되므로 for 문을 두 번 사용하는 것으로는 시간 복잡도를 줄일 수 없다. 따라서 이때는 알고리즘 자체를 수정해야한다.
- `for(int i = 0; i < strlen(s) ; i ++;)` 이라는 문장이 있다면 이는 $O(n^2)$ 이다. (단, 문자열 s 의 길이가 n 이다.) `strlen(s)`라는 함수 또한 n 만큼 연산을 수행함. 따라서 반복문은 그 자체로 n 회의 연산을 하기때문에 반복문을 최소로 사용한다면 시간 복잡도를 줄이는데 도움이 될 것이다.
- 만약 n 개의 데이터를 가져와서 배열에 할당한다고 한다면 그것또한 $O(n)$ 이다.

다음은 1 초의 시간 제한에서, 문제들에서 자주 볼 수 있는 제한과 의도된 풀이의 시간 복잡도의 예시들입니다.

- $n \leq [10^9, 10^{18}]$: $O(1)$, $O(\log n)$, $O(\log 2n)$ 등이 정해진 문제들입니다. 오래된 문제들에서는 10^9 이 많이 사용되었지만 계산이 매우 빠르게 가능한 경우 $O(n)$ 이 통과될 수 있는 경우가 많아 최근에는 대부분 8 바이트 long long 범위를 적극 활용하여 $n \leq 10^{18}$ 로 사용됩니다. 간혹 $O(\sqrt{n})$ 와 같이 특이한 시간 복잡도를 원하는 경우 $n \leq 10^{12}$ 와 같은 제한을 사용하기도 합니다.
- $n \leq [10^5, 10^6]$: $O(n \log n)$ 의 시간 복잡도를 요구하는 문제에 사용됩니다. 오래된 문제들의 경우 주로 $n \leq 10^5$ 를 많이 사용했으나, 최근에는 상수가

작은 $O(n^2)$ 풀이가 존재하는 경우 1 초 내에 통과되는 경우가 많아 $n \leq 2 \cdot 10^5$ 으로 많이 사용하는 추세이고, 전체적으로 정해의 연산이 가볍고 타이트한 제한을 사용하려는 경우 $n \leq 5 \cdot 10^5$ 이나 $n \leq 10^6$ 까지도 사용됩니다. 로그가 붙는 것을 허용하지 않게 완전히 $O(n)$ 풀이만을 원하는 경우 $n \leq 10^8$ 정도의 제한이 사용되는 경우도 있습니다.

- $n \leq [1000, 10000]$: 주로 $O(n^2)$ 을 요구하는 문제에 사용됩니다. 이 제한 역시 이전에는 $n \leq 1000$ 으로 사용한 경우가 많았지만 n^3 이 10 억밖에 되지 않기 때문에 $O(n^3)$ 풀이가 통과되는 경우가 아주 많아 최근에는 대체로 최소 $n \leq 5000$ 이상이 많이 쓰이고 있습니다. $n \leq 1000$ 이나 $n \leq 2000$ 은 $O(n^2 \log n)$ 를 요구하는 문제에도 종종 사용됩니다.
- $n, m \leq [1000, 10000]$: 위와 비슷하게 $O(nm)$ 정도의 시간 복잡도를 요구하는 경우에 많이 사용됩니다. 다만, 격자판의 상태를 입력으로 주는 것과 같이 입력량도 $O(nm)$ 인 문제에서는 입력 속도를 고려해서 최대 $n, m \leq 3000$ 정도가 제한으로 많이 사용됩니다.
- $n \leq [200, 500]$: $O(n^3)$ 을 요구하는 문제에 많이 사용됩니다.
- $n \leq [10, 20]$: $O(2^n)$, $O(n \cdot 2^n)$, $O(n!)$ 등의 지수 이상 시간 복잡도를 요구하는 문제에 주로 사용됩니다. 특히 $O(n!)$ 의 경우에는 매우 빠르게 증가하기 때문에 n 은 거의 10 이하로 제한을 걸게 됩니다.

출처 : <https://www.secmem.org/blog/2021/01/22/time-complexity/>