

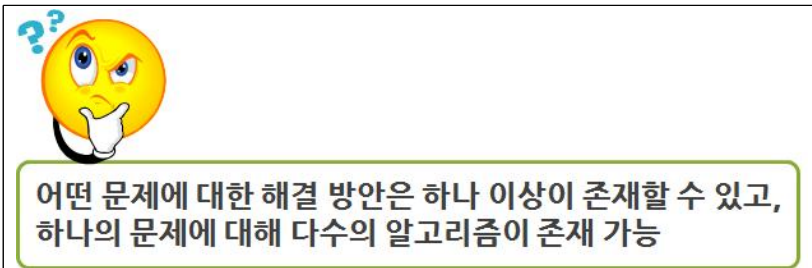
2주차 3차시 성능분석

【학습목표】

1. 알고리즘의 선택 기준과 성능 분석 방법을 설명할 수 있다.
2. 시간 복잡도 표기법의 사용 및 순서를 설명할 수 있다.

학습내용1 : 알고리즘의 선택기준

1. 개요



2. 알고리즘 분석 기준

<가장 최적의 알고리즘 결정을 위한 알고리즘 분석, 평가 필요>

* 판단기준 : 정확성, 명확성, 수행량, 메모리 사용량, 최적성 등

- ① 정확성 : 올바른 입력에 대하여 유한 시간에 올바른 결과를 출력하느냐
- ② 명확성 : 알고리즘의 표현이 얼마나 이해하기 쉽고 명확하게 작성되었느냐
- ③ 수행량 : 기본적인 연산을 제외한 알고리즘 특성을 나타내는 연산자 분석
- ④ 메모리 사용량 : 알고리즘에 쓰이는 메모리 사용량 분석
- ⑤ 최적성 : 가장 중요 기준으로 알고리즘 적용 시스템 환경에 따라 수행량, 메모리 사용량이 달라지기 때문

가장 좋은' 알고리즘이란 '최적의' 알고리즘

학습내용2 : 알고리즘의 성능분석 방법

1. 알고리즘 성능 분석 방법

① 공간 복잡도

- 알고리즘을 프로그램으로 실행하여 완료하기까지 필요한 총 저장 공간의 양
- 공간 복잡도 = 고정 공간 + 가변 공간



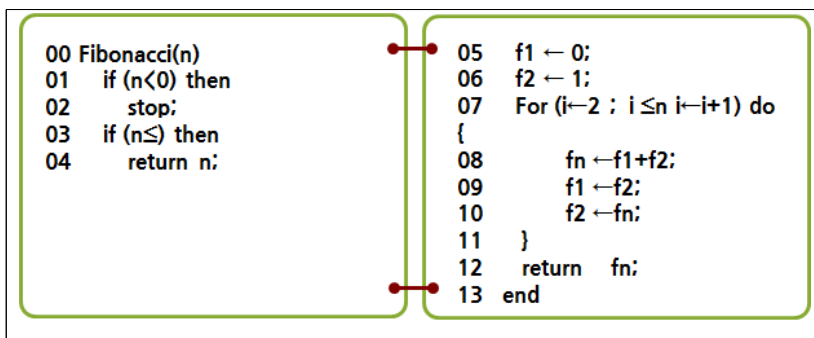
② 시간 복잡도

- 알고리즘을 프로그램으로 실행하여 완료하기까지의 총 소요시간
- 시간 복잡도 = 컴파일 시간 + 실행 시간



* 실행 빈도수의 계산 : 지정문, 조건문, 반복문 내의 제어문과 반환문은 실행시간 차이가 거의 없으므로 하나의 단위시간을 갖는 기본 명령문으로 취급

* 예 : 피보나치 수열 알고리즘의 빈도수 구하기



- $n < 0$, $n = 0$, $n = 1$ 의 경우에 대한 실행 빈도수
- for 반복문이 수행되지 않기 때문에 실행 빈도수가 작음

피보나치 수열 알고리즘의 실행 빈도수 : $n < 0$, $n = 0$, $n = 1$ 의 경우			
행 번호	$n < 0$	$n = 0$	$n = 1$
1	1	1	1
2	1	0	0
3	0	1	1
4	0	1	1
5~13	0	0	0

- $n > 1$ 의 일반적인 경우에 대한 실행 빈도수
- n 에 따라 for 반복문 수행

피보나치 수열 알고리즘의 실행 빈도수 : $n > 1$ 의 경우			
행 번호	실행빈도수	행 번호	실행빈도수
1	1	8	$n-1$
2	0	9	$n-1$
3	1	10	$n-1$
4	0	11	0
5	1	12	1
6	1	13	0
7	n		

- 총 실행 빈도수 = $1+0+1+0+1+1+n+(n-1)+(n-1)+(n-1)+0+1+0 = 4n+2$

1) 시간 복잡도 표기법

<빅-오(Big-Oh) 표기법 사용>

- 빅-오(Big-Oh) 표기법 순서
 - 실행 빈도수를 구하여 실행시간 함수 찾기
 - 실행시간 함수의 값에 가장 큰 영향을 주는 n 에 대한 항을 선택함
 - 계수는 생략하고 O (Big-Oh)의 오른쪽 괄호 안에 표시
- 피보나치 수열의 시간 복잡도 = $O(n)$
 - 실행시간 함수 : $4n+2$
 - n 에 대한 항을 선택 : $4n$
 - 계수 4는 생략하고 O (Big-Oh)의 오른쪽 괄호 안에 표시 : $O(n)$

2) 각 실행 시간 함수에서 n 값의 변화에 따른 실행 빈도수 비교

$\log n$	<	n	<	$n \log n$	<	n^2	<	n^3	<	2^n
0		1		0		1		1		2
1		2		2		4		8		4
2		4		8		16		64		16
3		8		24		64		512		256
4		16		64		256		4096		65536
5		32		160		1024		32768		4294967296

【학습정리】

1. 알고리즘 성능 분석은 실행에 필요한 공간 측면에서 분석하는 공간 복잡도와 실행에 소요되는 시간 측면에서 분석하는 시간 복잡도를 추정하여 일반적인 평가를 한다.