

## 자료구조 2주차 (성능 분석) 퀴즈

제출: 2023년 3월 13일

제출자: 민경하

풀이: 김동주

문항	문제
1	특정한 프로그램을 수행하는 데 요구되는 메모리의 크기를 [ ](이)라고 한다. <i>공간 복잡도</i>
2	점근적 분석법에서 성능은 [ ]에 따라서 결정된다. <i>입력의 크기</i>
3	If $g(n) \geq M f(n)$ for $n > n_0$ , then $g(n) = [ ]$ .에서 [ ]에 들어갈 식은? <i><math>O(n)</math></i>
4	다음 네 가지 표현 중에서 그 의미가 다른 것은 무엇인가? (1) $g(n)$ 은 $f(n)$ 보다 성능이 나쁨 <i><math>f(n) &lt; g(n)</math></i> (2) 최악의 경우에도 $f(n)$ 은 $g(n)$ 보다 좋음 <i><math>f(n) &lt; g(n)</math></i> (3) $f(n)$ 의 하한은 $g(n)$ 임 <i><math>f(n) \geq g(n)</math></i> (4) $f(n) \leq g(n)$
5	$f(n) = O(5n \log n)$ 라면, $f(n) = O(g(n))$ 이다.에서 $g(n)$ 에 올 수 <u>없는</u> 함수는? <i><math>O(n \log n)</math></i> (1) $n$ (2) $n^2$ (3) $n^3$ (4) $n \log n$
6	$f(n) = O(n^2 + n)$ 라면, $f(n) = O(g(n))$ 이다.에서 $g(n)$ 에 올 수 <u>없는</u> 함수는? <i><math>O(n^2)</math>?</i> (1) $n$ (2) $n^2$ (3) $n^3$ <i><math>\rightarrow n^3</math></i>

	<p>✓(4) <math>n^4 \rightarrow n^4</math>?</p>
7	<p>n 명인 반의 학생들이 모두 키 순으로 앉아있다. 여기에 새로운 학생 한 명이 전학을 왔다. 이 학생을 그냥 맨 앞에 앉도록 할 때, 걸리는 시간 복잡도는 무엇인가?</p> <p>(1) <math>O(1)</math>  (2) <math>O(n)</math>  (3) <math>O(\log n)</math>  (4) <math>O(n \log n)</math></p>
8	<p>다음 함수의 시간 복잡도를 n의 함수로 나타내시오.</p> <pre>int func ( int n ) {     for ( k = 1; k &lt; n; <u>k = k * 10</u> )         printf("hello"); }</pre> <p><math>O(\log n)</math></p> <p>(1) <math>O(n)</math>  (2) <math>O(n^2)</math>  (3) <math>O(\log n)</math>  (4) <math>O(n \log n)</math></p>
9	<p>다음 중 시간 복잡도가 가장 큰 함수는 무엇인가?</p> <p>(1) <math>O(n^{100})</math>  (2) <math>O(2^n)</math>  (3) <math>O(n!)</math>  (4) <math>O(n^n)</math></p> <p> <math>\rightarrow 2^n &gt; n^{100} \ (n &gt; ?) \rightarrow 2^n ? n^{100}</math>  <math>\rightarrow n! &gt; 2^n \ (n &gt; 3)</math>  <math>\rightarrow n^n \geq n!</math> </p> <p> <math>n \log 2 ? 100 \log n</math>  <math>n &gt; \log n</math> 이므로 <math>2^n &gt; n^{100}</math> 임.  충분히 큰 n에서는 </p>
10	<p>다음의 네 가지 시간 복잡도 중에서 가장 시간이 많이 걸리는 시간 복잡도는 무엇인가?</p> <p>(1) <math>O(n!)</math>  (2) <math>O(n^n)</math>  (3) <math>O(2^n)</math>  (4) <math>O(n^{100})</math></p> <p> <math>\leftarrow n</math>이 충분히 클 때  <math>\leftarrow n=1</math>일 때 (?)  <math>\leftarrow n</math>이 100 미만일 때 </p>
11	<p>다음의 네 가지 시간 복잡도 중에서 가장 시간이 적게 걸리는 시간 복잡도는 무엇인가?</p> <p>(1) <math>O(n!)</math></p> <p>충분히 큰 n에서 값을 가정</p> <p><math>\leftarrow n \leq 3</math>일 때</p>

n의 값에 따라  
정답이 상이하나,

Big-O Notation은

Worst-case일 때의

시간 복잡도를 나타내므로

$n \rightarrow \infty$  일 때 가정하고

풀이 함.

	(2) $O(n^n)$ (3) $O(2^n)$ (4) $O(n^{100})$ ← $n$ 이 매우 클 때.
12	다음 연산들 중에서 가장 시간이 적게 걸리는 시간 복잡도의 연산은 무엇인가? (1) 정렬되지 않은 배열에서 삽입 $O(1)$ ? → 삽입 후 정렬 요+α = 랜덤 액세스 (2) 정렬되지 않은 연결 리스트에서 삭제 $\Omega(1) \sim O(n)$ (3) 정렬된 배열에서 이진 탐색 (binary search) $\Omega(1) \sim O(\log n)$ (4) 버블 정렬 $O(n^2)$
13	다음 연산들 중에서 가장 시간이 많이 걸리는 시간 복잡도의 연산은 무엇인가? (1) 정렬되지 않은 배열에서 삽입 (2) 정렬되지 않은 연결 리스트에서 삭제 (3) 정렬된 배열에서 이진 탐색 (binary search) (4) 버블 정렬
14	성능을 측정할 때 <u>보장</u> 의 의미를 부여하기 위해서 선택하는 성능은 다음 중 무엇인가? → worst case (1) 최선의 경우의 성능 (2) 평균의 경우의 성능 (3) 최악의 경우의 성능 (4) 최초의 경우의 성능
15	$f(n) = O(n^2)$ 에 대한 설명으로 맞지 않은 것은? (1) $f(n)$ 은 어떤 경우에도 $n^3$ 보다 빠르다. ○ (2) $f(n)$ 의 최악의 경우는 $n^2$ 이다. ○ (3) $f(n)$ 의 상한은 $n^2$ 이다. ○ (4) $f(n)$ 은 $n^2$ 보다 성능이 더 나쁘다. ✗