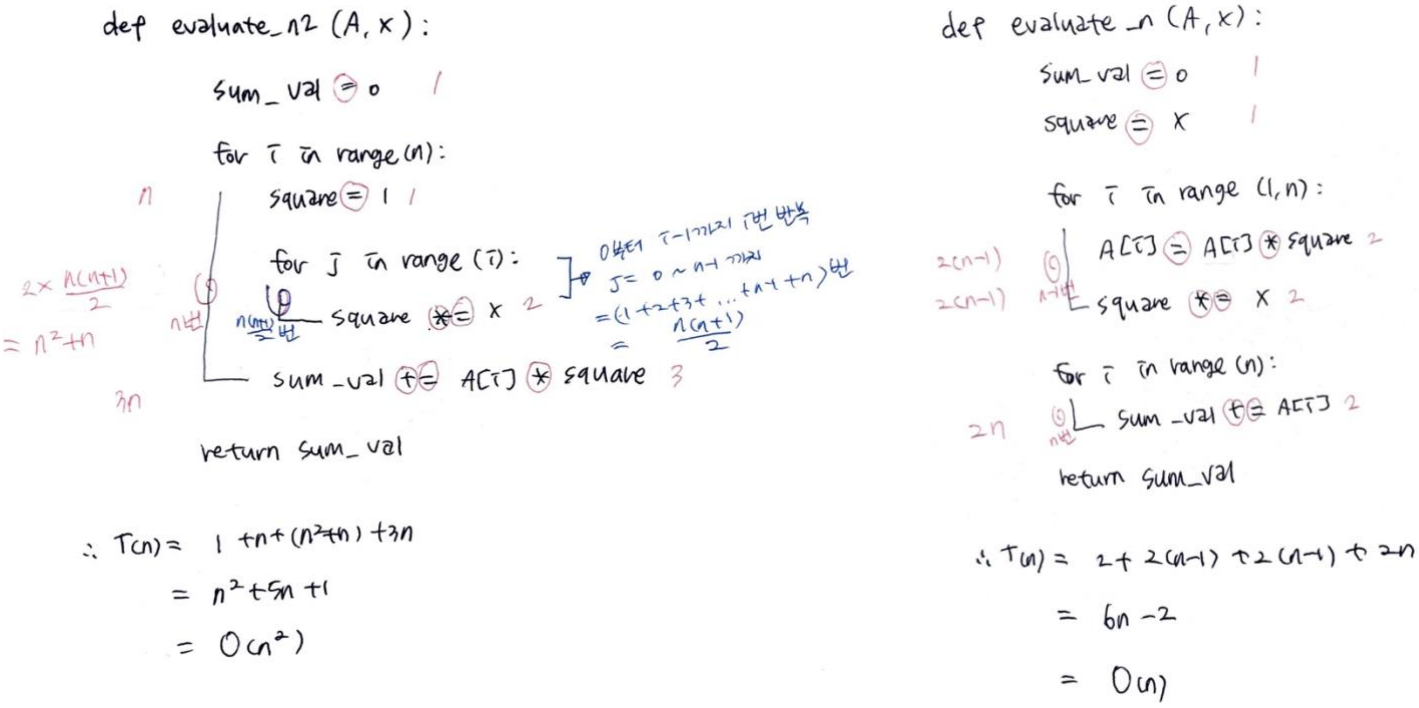
[알고리즘]

과제 #1. 다항식 수행 시간 비교 및 분석

Global Business & Technology
201904385 우인경

1. 수행 시간 분석



크기가 n인 임의의 리스트 A에 대해 주어진 함수의 값을 구하는 코드를 두 가지 방법으로 구현하였다. 첫번째 방법은 이중for문을 사용하여 i가 증가할 때 마다 그에 따른 x의 i제곱 값을 계산하고 A[i]와 곱한 값을 계산하여 더하는 방법으로 이 때의 시간복잡도 T(n) = O(n^2)이다.

두번째 방법은 단일for문을 사용하여 i가 증가할 때 리스트의 i번째 요소인 A[i]부터 마지막 요소까지 x를 곱해 a[i] * x^(i)가 되는 리스트를 먼저 구하고 그 합을 더하여 계산하는 방법이다. 이 때의 시간복잡도 T(n) = O(n)이다.

2. 실제 실행 시간 비교

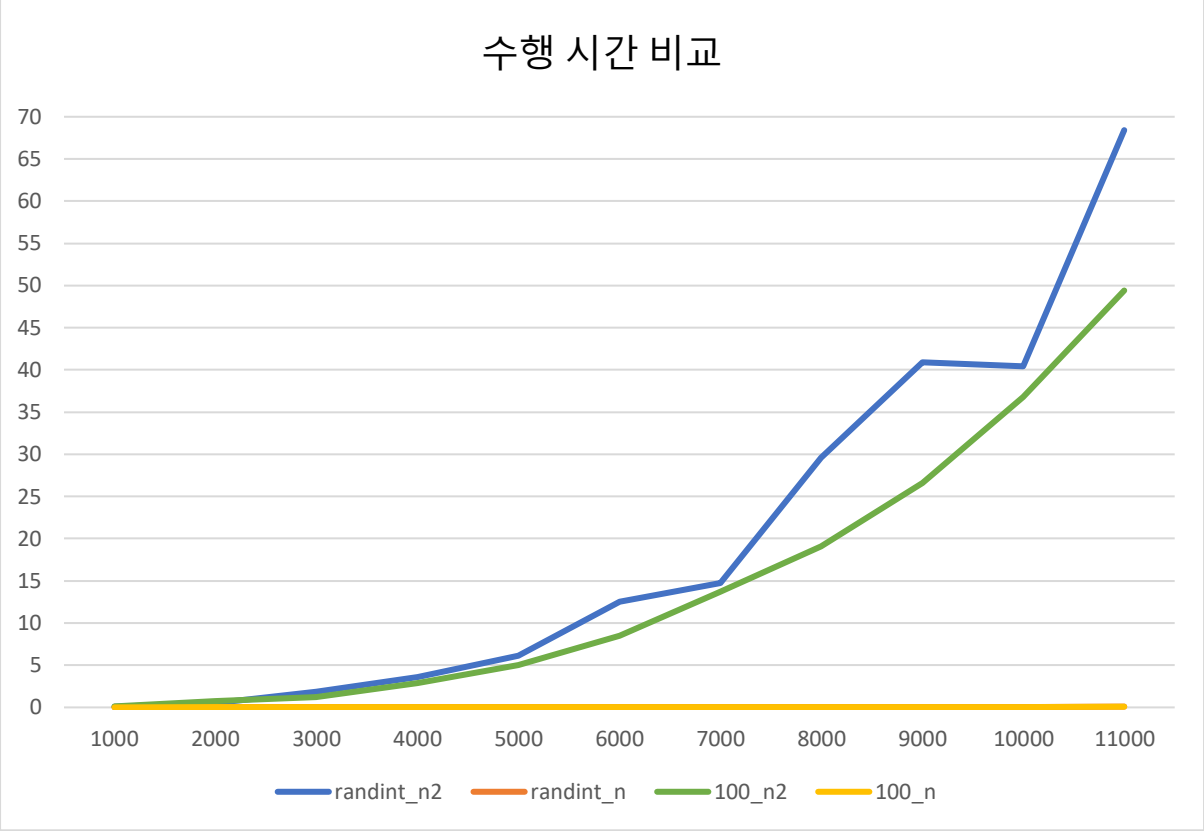
제한 시간인 60초 내에서 실행되는 n을 1,000부터 11,000까지 입력하며 이에 따른 수행 시간을 측정해보았다. x값은 randint 함수를 사용해 임의의 값을 입력 받았으며 두 함수의 입력 크기(n)에 따른 실제 실행 시간은 다음과 같다.

	x = randint	-147	104	783	391	-401	-815	128	-705	875	-160	-566
	입력크기 n	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	11000
수행 시간	evaluate_n2	0.081248	0.495329	1.863496	3.572315	6.11661	12.50486	14.75811	29.58625	40.89181	40.44487	68.41306
	evaluate_n	0.001261	0.002729	0.007863	0.012635	0.018453	0.030727	0.043009	0.068837	0.06849	0.059744	0.094503

동일한 환경에서의 테스트를 위해 x를 100으로 고정하고 다시 측정해본 결과는 다음과 같다.

x = 100												
입력크기 n		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	11000
수행 시간	evaluate_n2	0.119484	0.730426	1.261849	2.900207	5.031379	8.510455	13.67367	19.12098	26.61913	36.78949	49.40859
	evaluate_n	0.001262	0.004126	0.005753	0.012142	0.018758	0.028271	0.040385	0.035514	0.044947	0.054626	0.08059

표의 결과를 파악하기 쉽게 그래프로 나타내었다. randint_n2와 randint_n 는 임의의 x값을 입력받은 결과이고 100_n2와 100_n은 x를 100으로 동일하게 고정하여 측정한 결과이다. 전반적으로 x값이 다름에도 비슷하게 수행 시간이 증가하는 양상을 보인다. N이 1000~5000인 구간에서는 차이가 크지 않다가 점차 n이 증가함에 따라 수행시간 역시 급격하게 증가하며 차이가 커진다. 이를 통해 evaluate_n2에 비해 evaluate_n의 수행시간이 훨씬 짧은 것을 알 수 있다.



3. 느낀점

같은 값을 출력하는 함수라도 어떻게 구현하는지에 따라 수행 시간이 달라지는 것을 알 수 있다. 과제에서 구현한 함수는 비교적 간단한 함수이기 때문에 차이가 극적으로 크지는 않지만 코드가 길고 복잡해질 수록, 그리고 입력 값이 커질 수록 차이도 커질 것이기 때문에 반복문의 중첩을 최소화하고 코드의 효율을 높여 좋은 알고리즘을 구현하는 것이 중요하다.