**2주차 실습문제 문제 3-2 실행시간 비교 보고서**

**20011818 정소윤**

1. **텍스트이(가) 표시된 사진

   자동 생성된 설명텍스트이(가) 표시된 사진

   자동 생성된 설명소스코드**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

메모리 동적할당을 main함수 내에서 한 다음, 그 메모리 주소를 함수로 넘기면서 동적할당의 시간은 실행시간 측정에 포함하지 않으려고 했습니다.

1. **실행시간 비교**
2. 배열 크기 : 10

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 배열 크기 : 100

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 배열 크기 : 500

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 배열 크기 : 1000

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 배열 크기 : 5000

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 배열 크기 : 10000

텍스트이(가) 표시된 사진

자동 생성된 설명

**실행시간 비교 표**

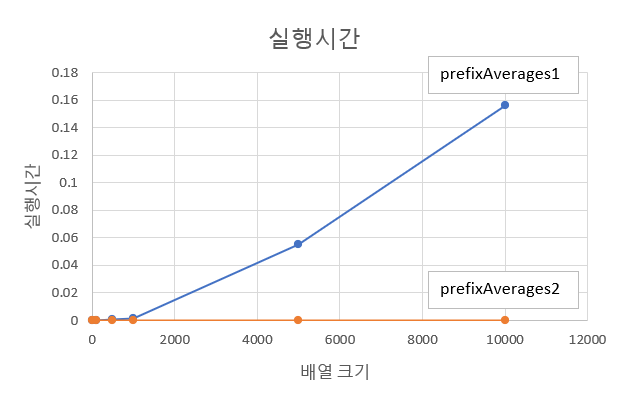
테이블이(가) 표시된 사진

자동 생성된 설명

편의상 표에서는 prefixAverages1 은 ‘pA1’으로 prefixAverages2는 ‘pA2’로 표기하였습니다. 배열 크기는 10, 100, 500, 1000, 5000, 10000 6개의 경우로 각 5번씩 실행하였다.

표에서 볼 수 있듯이 어떤 입력 크기이든지 prefixAverages2가 prefixAverages1보다 실행시간이 덜 걸린다는 것을 알 수 있다.

배열의 크기에 따른 평균 실행시간 추세를 살펴보면 prefixAverages1, prefixAverages2 모두 입력 크기가 증가함에 따라 프로그램을 실행하는 시간도 증가함을 알 수 있다.



입력 크기가 작을 때는 그 차이가 별로 크지 않았는데 입력 크기가 커질수록 둘 사이의 차이가 커진다. 위 그래프를 보면 prefixAverages1의 실행시간 증가율이 prefixAverages2의 실행시간 증가율보다 훨씬 크다는 것을 알 수 있다. 2주차 강의에서 배웠던 것처럼 점근적 방식(Big-Oh)으로 각 방식의 실행시간을 측정하였을 때도 prefixAverages1의 실행시간은 O(n²)이고 prefixAverages2의 실행시간은 O(n)이기에 prefixAverages1의 실행시간이 입력크기 n이 커짐에 따라 더 크게 증가한다.