

# Homework3: Stack with Dynamic Array vs. Linked List

Data Structures, Konkuk Univ.

2021. 3. 25 (목)

수업 시간에 배운, stack을 dynamic array와 linked list를 이용하여 구현하시오. Dynamic array의 경우, 2배 증가(초기 크기=1) 버전을 구현하여, 아래의 조건으로 실험을 수행할 것.

- (1) (a) Push를 100만번 한 후, pop을 100만번하는 실험, (b) Push를 200만번 한 후, pop을 200만번하는 실험, ... (j) Push를 1000만번 한 후, pop을 1000만번하는 실험을 dynamic array와 linked list에서 각각 수행하여 수행시간을 비교하시오. (Push하는 숫자는 0부터 시작해서 1씩 증가하며 push하시오.)
- (2) (1-j)번 실험을 변경해서, dynamic array와 linked list의 push와 pop의 1회 수행 시간에 대한 time complexity를 측정할 수 있도록 실험을 설계해서 결론을 제시하시오. 보고서에 실험을 어떻게 설계했는지, 그래서 결론은 수업시간에 배운 time complexity와 비교해서 어떠한 결과를 나왔는지를 설명하시오.
- (3) Dynamic array를 이용하는 경우, array를 2배로 증가시킬 때, 수행시간이 다른 경우에 비해 증가하는 것을 실험으로 측정하여 보이시오. Array가 2배로 증가할 때 수행시간이 증가하는 것을 한 구간 정도 보이면 됩니다. 예를 들면,  $n=2^{20}$ 에서 array가 2배로 증가된다면,  $n=2^{20}-10$ 부터  $n=2^{20}+10$ 정도 구간에서 한번의 insert operation들이 몇 millisecond가 걸리는지 측정해서 보이시면 됩니다.
- (4) Push를 1번, pop을 1번씩 교대로 (a) 100만번, (b) 200만번, ..., (j) 1000만번 수행하는 실험을 하여 실행 시간을 비교하시오. 즉, push(0), pop(), push(1), pop(), push(2), pop(), push(3), pop()... 을 수행할 것. 수행 시간에 대한 간단한 결론을 (1)번 실험과 비교하여 제시하시오. (1)번 실험과 (4)번 실험은 우리가 stack 사용하는 두 극단적인 예를 실험한 것입니다. 구체적으로 어떻게 극단적인 예인지 설명하고, 일반적인 사용은 어떤 식으로 stack을 접근하는 것이라고 생각하나요?

## 제출방법:

- (1) 소스 파일(**\*\*\*.c** 또는 **\*\*\*.cpp**)과 보고서를 ecampus에 제출하시오. 소스 파일 맨 윗쪽에 코멘트로 학번과 이름, 실행환경(Windows, Mac, Linux)을 명시할 것. **작성한 function들이 몇 번 문제에 해당하는 function인지 대한 설명을 코멘트를 반드시 남기고, 수행시간을 printf할 때도 어떤 문제의 수행시간인지를 구별할 수 있도록 하시오.**

제출기한: 2021.4.1(목) 23:59:59 이캠퍼스 시간 기준 (마감 시간 이후는 절대 불인정)