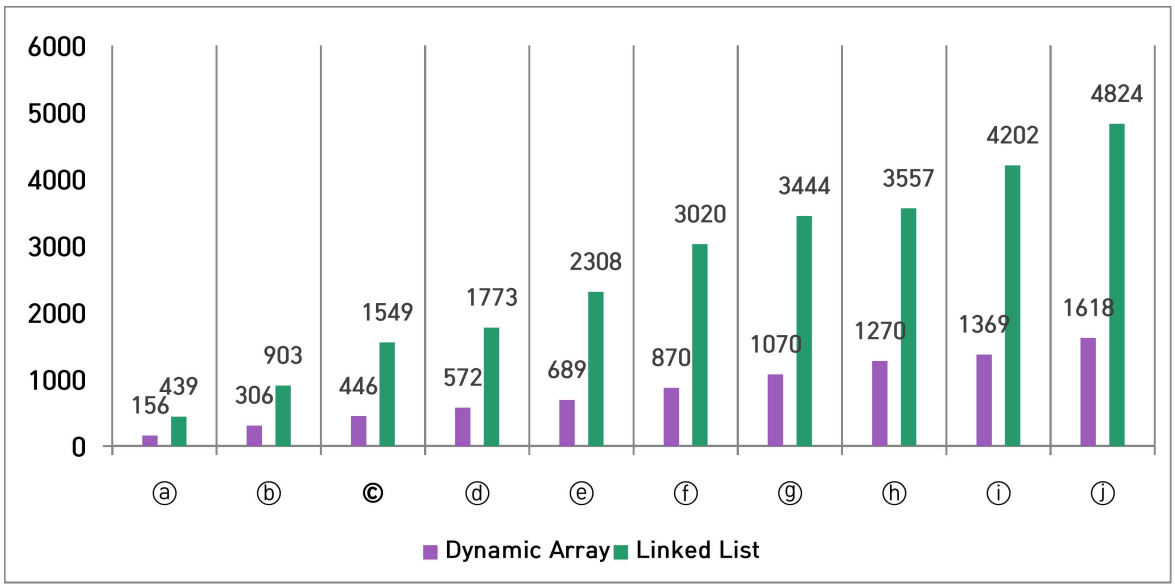
Homework3: Stack with Dynamic Array vs. Linked List
Data Structures, Konkuk Univ.

201811281 이유준

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.)

Index	ⓐ	ⓑ	©	ⓓ	ⓔ	ⓕ	ⓖ	ⓗ	ⓘ	ⓙ
Dynamic Array	156	306	446	572	689	870	1070	1270	1369	1618
Linked List	439	903	1549	1773	2308	3020	3444	3557	4202	4824



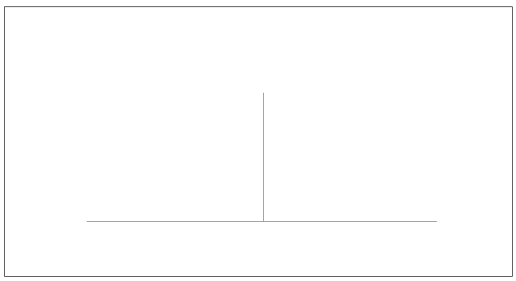
2. (1-j)번 실험을 변경해서, dynamic array와 linked list의 push와 pop의 1회 수행 시간에 대한 time complexity를 측정할 수 있도록 실험을 설계해서 결론을 제시하시오. 보고서에 실험을 어떻게 설계했는지, 그래서 결론은 수업시간에 배운 time complexity와 비교해서 어떠한 결과를 나왔는지를 설명하시오.

Index	0 ~ 1million	1million ~ 2million	2million ~ 3million	3million ~ 4million	4million ~ 5million	5million ~ 6million	6million ~ 7million	7million ~ 8million	8million ~ 9million	9million ~ 10million
Dynamic Array	0.000087	0.000082	0.000078	0.000075	0.000085	0.000075	0.000075	0.000075	0.000090	0.000090
	0.000075	0.000076	0.000075	0.000076	0.000073	0.000076	0.000076	0.000076	0.000075	0.000077
Linked List	0.000222	0.000231	0.000235	0.000224	0.000233	0.000224	0.000223	0.000227	0.000229	0.000229
	0.000250	0.000239	0.000241	0.000240	0.000237	0.000242	0.000239	0.000232	0.000239	0.000233

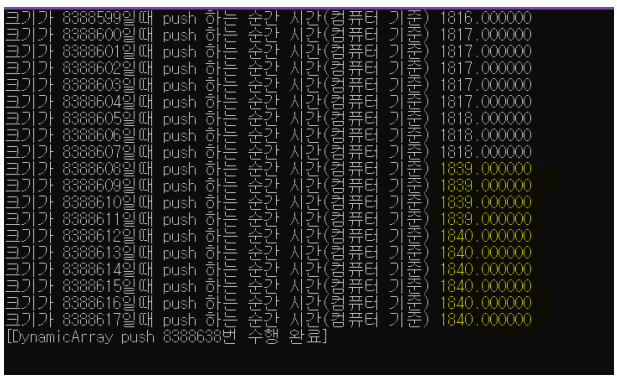
변화량이 일정한 시간에서 + 또는 - 해서, 둘 다 O(1)이라고 생각이 듭니다.

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=2^20에서 array가 2배로 증가된다면, n=2^20-10부터 n=2^20+10 정도 구간에서 한번의 insert operation들이 몇 millisecond가 걸리는지 측정해서 보이면 됩니다.

저는 수행 할 범위를 2^23-10 ~ 2^23 +10으로 잡고 각 수행 시간 중 수행 시간이 급격히 증가하는 구간이 생기고, 그 구간은 Dynamic Array의 Stack Double 이 수행되는 부분 임을 코드로 표현했습니다. (배열 크기가 2배 증가하는 부분인 2^23인 부분에서 시간이 급격히 증가하는 것을 표현)



와 같은 배열 크기를 두배 증가할 때만 시간이 좀 오래 걸리는 결과값을 원했으나,
배열의 크기가 2배 증가하고 나서, 계단식 모양의 결과값이 나왔습니다.



배열의 크기가 달라지고 나서, 걸리는 순간 시간이 달라지는 결과를 보였습니다.

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?
1번보다 좀 적은 시간이 걸리긴 했지만, 전반적으로 무슨 차이인지는 잘 모르겠습니다.
단지 1번은 장거리 여행을 가는 여행 수단 같은 느낌이라면 4번은 단거리 이동을 가는 이동수단을 표현한 느낌이었습니다. 일반적인 사용은 stack을 4번처럼이라 느낍니다. 1번과 같은 장거리 여행 이동 느낌은 FIFO나 LIFO나 큰 차이가 없을 것이지만, 단거리 여행은 LIFO와 FIFO가 큰 차이가 있을 것이고 이는 Stack의 활용이 큰 도움이 될 것이라 여깁니다.

Index	ⓐ	ⓑ	ⓒ	ⓓ	ⓔ	ⓕ	ⓖ	ⓗ	ⓘ	ⓙ
Dynamic Array	108	242	285	442	484	681	807	887	939	971
Linked List	439	992	1354	1774	2159	2587	2738	3144	3605	4096