Homework 2: Array vs. Linked List

Data Structures, Konkuk Univ.

2021. 3. 18 (목)

1. **수업 시간에 배운, array와 singly linked list를 구현하시오.** Linked list의 경우, 수업 슬라이드에 나오는 예제에서는 첫 번째 element의 index를 1로 가정하지만, 이 숙제에서는 **일반적인 C언어 style대로 0으로 가정하는 것을 주의할 것**. c프로그램에 대한 시간 측정은 첨부된 예제 프로그램을 참고하여 작성. 수행시간의 경우 millisec 단위의 int로 표시하시오.

(1) **Insertion**: array의 경우는 integer type으로 100,000개를 할당해서, i번째 element인 array[i]에 i를 assign하는 문장을 i가 0에서 99,999까지 실행시키는 시간을 출력하는 프로그램을 작성하시오. 즉, 100,000개의 data를 array 끝에 shift 없이 insert하는 시간입니다. Linked list의 경우는 head쪽으로 insert를 100,000번 수행하되 data는 0에서부터 99,999까지 차례대로 입력하고, 수행시간을 출력하는 프로그램을 작성하시오.

(2) **Random access for read**: Array와 linked list 모두, random한 index를 발생시켜서, 그 index에 있는 data값을 계속해서 더해가는(100,000번 반복) program을 작성하시오. (더한 값을 저장하는 변수인 sum의 최대치를 고려해서 type을 선언할 것)

(3) **Random access for deletion**: Array와 linked list 모두, random한 index를 발생시켜서, 그 index에 있는 data를 삭제(100,000번 반복)하는 program을 작성하시오. **특히, Array의 경우, 해당 data를 삭제하고 shift하는 것까지 구현할 것.** 주의: 삭제할 data의 index를 발생시킬때 처음에는 0~99,999, 그 다음에는 개수가 99,999이므로, random index 0~99,998, ... , 이런 식으로 범위를 줄이면서 반복하여 합니다.

주의: c언어의 rand() 함수는 0~RAND_MAX (즉, 32767) 중에 임의의 수를 return해 줍니다. 그러나, 이 문제에서는 0~(current_size-1)까지의 임의의 수를 발생시켜야 하므로, 다음처럼 구현해야 전체 범위를 access할 수 있습니다.

```
index = rand() * 10 % current_size;
```

2. **각 operation들의 time complexity 측정**

위에서 작성한 프로그램들은 data structure의 크기(n)가 1에서 100,000으로 커짐에 따라, 각 operation의 수행시간을 n=1일 때부터 n=100,000까지의 시간을 모두 더한 것입니다. 이 문제에서는 각 operation의 1회 수행시간이 n에 따라서 어떻게 변화하는지를 측정하려고 합니다. 예를 들면, insert operation을 array 크기가 10000일 때 수행하는 것과 20000일 때, 30000일 때, ..., 100000일때 수행하는 것들의 수행시간 차이가 있는지를 측정하고자 합니다. Array와 linked list의 경우, insert는 time complexity가 O(1)이기 때문에, 수행시간이 n에 따라서 대략 일정할 것입니다. 하지만, delete operation의 경우는 array와 linked list가 모두 O(n)이기 때문에, n=10000일 때, 20000일 때, 30000일 때, ..., 100000일 때,

수행시간이 점점 linear하게 증가할 것입니다. 이렇게 **insert operation과 delete operation의 time complexity를 실제 측정**하고자 합니다.

1번 문제에서 측정한 결과, **각 operation의 1회 수행시간이 millisec 단위로 측정하기에 상대적으로 짧고 변이가 클 수 있어서**, 10,000번씩 묶어서 평균을 내어서 그래프를 그려보고자 합니다. 즉, 1번 문제에서의 수행시간을 1~10000, 10001~20000, 20001~30000, 30001~40000, 40001~50000, 50001~60000, 60001~70000, 70001~80000, 80001~90000, 90001~100000 구간으로 나누어 측정하고, 그 시간을 10,000으로 나누어, **(1) insert operation과 (2) delete operation의 1회 수행시간이 data structure의 크기에 따라 어떻게 증가하는지 표와 그래프로 그려서, 수업시간에 배운 time complexity를 고려해서 설명하는 보고서를 작성하시오.**

제출방법:

(1) **소스 파일**(***.c 또는 ***.cpp)과 **보고서**를 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: 2021.3.25(목) 23:59:59 이캠퍼스 시간 기준 (마감 시간 이후는 절대 불인정)