[GBC 알고리즘] 과제 1-1

2020. 12. 28.

21900346 백승준

1. vector, list, int형 배열 세 개의 차이를 설명해주세요.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Standard Template Library | | Array |
| 예시 | vector<int> vec; | list<int> lst; | int arr[10]; |
| 특징 | -자동으로 메모리 할당  -연속적 메모리 할당 | -자동으로 메모리 할당  -Linked list 방식 | 변수 선언 시에 크기를 할당해야 한다. |
| 장점 | 특정 인덱스 직접 접근 가능 | 컨테이너의 어느 위치에서라도 삽입/제거가 빠름 | 특정 인덱스 직접 접근 가능 |
| 단점 | 컨테이너의 끝 위치가 아닌 위치에서 삽입/제거 시 그 성능이 list보다 현저히 느림 | 특정 인덱스 직접 접근 불가 (선형 탐색) | -선언 시에 크기를 할당해야만 함(overflow)  -선언 후에 중간에 배열 크기 변경 불가능 |

1. vector, list, set, map를 대충 설명하고, 각각에 대해 int type으로 해서 값 삽입, 삭제, 수정, 출력(현재 저장된 값들 전부 출력)하는 코드를 캡처해주세요. (메소드를 사용해서 값을 수정,삽입하는 방법과 배열 사용하듯이 하는 방법이 있는데 편한 것으로 하세요) (출력은 반복문(for)을 사용해서 출력해주세요)

* vector: 자동으로 메모리가 할당되는 동적 배열이다.
* list: vector와 같이 자동으로 메모리가 할당되는 동적 배열이긴 하나 vector는 연속적인 메모리를 할당하는 반면에 list는 비연속적인 메모리를 할당한다.
* set: 균형 이진 탐색 트리 방식으로 구성되어 있으며 데이터 간의 구별을 위해 중복된 데이터의 삽입을 허용하지 않으며 자동으로 정렬된다.
* map: set과 거의 비슷하나 key와 data로 구분되어 있어 중복된 data가 있더라도 key만 다르다면 삽입이 가능하다는 특징이 있다. 정렬은 key를 기준으로 정렬한다.
* 삽입, 삭제, 수정 (출력) 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

* 삽입, 삭제, 수정 (출력) 코드

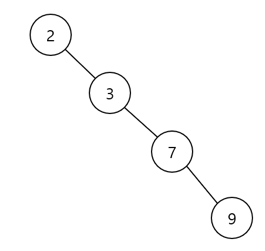
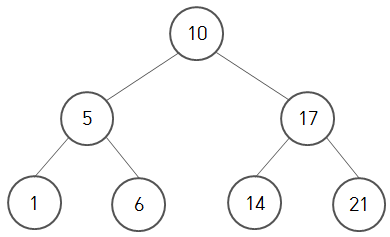
|  |
| --- |
|  |
|  |
|  |

1. set, map이 어떻게 내부가 구현되어 있는지 설명해주세요. (어떤 자료구조를 사용하는지)

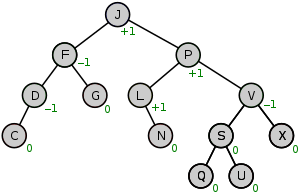
* set과 map은 균형 이진 탐색 트리를 사용한다.
* 균형 이진 탐색 트리를 이해하기 위해 이진 탐색 트리를 먼저 설명하자면

|  |
| --- |
| * 이진 탐색 트리는 다음과 같은 조건을 갖춘 데이터 구조이다.  1. 모든 노드의 key는 유일하다. 2. 왼쪽 서브트리의 key는 루트의 key보다 작다. 3. 오른쪽 서브트리의 key는 루트의 key보다 크다. 4. 모든 서브트리는 이진 탐색트리이다. |

* 이진 탐색 트리는 데이터 삽입 시 위치가 지정되어 검색 시 빠르게 검색할 수 있는 장점을 가지고 있지만 최악의 경우에는 그 장점을 갖추지 못한다.



* 왼쪽의 경우는 균형이 잘 잡힌 이진 탐색 트리이다. 이 경우에는 특정 데이터를 검색 시에 O(logn)의 시간이 소요된다. 하지만 오른쪽 경우에는 특정 데이터를 검색하기 위하여 O(n)의 시간이 소요되므로 이진 탐색 트리의 장점을 잘 살리지 못한다.
* 그래서 이러한 경우를 대비하기 위하여 균형 이진 탐색 트리가 등장하게 되는데 대표적으로 AVL 트리가 있다.
* AVL 트리는 이진 탐색 트리를 보완한 데이터 구조이므로 이진 탐색 트리의 특징을 기본적으로 갖추고 있다.
* 추가적인 특징은 모든 노드의 왼쪽 오른쪽 서브트리 높이 차이가 1 이하라는 점이다.



* 균형 이진 탐색 트리를 통해 이진 탐색 트리의 단점을 보완할 수 있게 되었으며 어떤 상황에도 검색 시 소요 되는 시간이 O(logn)을 갖출 수 있게 되었다.
* 그러므로 set과 map은 균형 이진 탐색 트리로 이루어져 있기 때문에 검색 시 빠르게 찾을 수 있다는 장점을 갖고 있다.

1. map에서 key값을 아직 넣지 않는 값으로 하면 뭐를 반환하나요? ( [key, value] 쌍에서 value가 int, double, string 일 때 각각에 대해 코드캡처)

**텍스트이(가) 표시된 사진

자동 생성된 설명**

텍스트이(가) 표시된 사진

자동 생성된 설명

1. sort, binary\_search, lower\_bound, upper\_bound 함수를 조사하고, int 배열, vector 에 대해 적절하게 사용해보세요 (코드캡처)

* sort: container의 값들을 오름차순으로 정렬하는 함수이다.
* binary\_search: 값들이 정렬되어 있다는 가정 하에서 이진 탐색으로 값을 찾는 함수이다.
* lower\_bound: 이진 탐색 기반으로, 값을 찾을 때 그 값보다 크거나 같은 첫번째 원소를 찾는 함수이다.
* upper\_bound: 값보다 큰 첫번째 원소를 찾는 함수이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* 코드

|  |
| --- |
|  |
|  |
|  |

1. numeric\_limits 라는 것을 이용해 int와 long long의 최대, 최소값을 출력해보세요. (코드 캡처)

**텍스트이(가) 표시된 사진

자동 생성된 설명**

텍스트이(가) 표시된 사진

자동 생성된 설명

1. int형(4바이트)의 값 범위가 얼마인지 억 단위 아래로 버림해서 적으세요 (대충 "파이는 3.14" 정도로 외우기 쉽게)

* -21억~21억

1. int& 타입의 특징이 뭔지 조사해서 적절히 사용한 예시코드 하나 작성해주세요 (코드 캡처)

* int& 타입을 Reference(참조자)라 부른다.
* 포인터와 같이 어떤 변수나 상수를 가리키는 역할을 하지만 포인터와는 다르다.
* 쉽게 말하자면 가리키는 변수의 별명이라고 말할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* 함수 인자로 레퍼런스를 사용하면 편리하다.