● Dynamic Array는 어떤 자료 구조인가?

▶ Dynamic Array는 size를 자동적으로 resizing하는 Array로, Array의 특징중에 fixed-size의 한계점을 보완하고자 고안된 자료구조입니다. Static Array의 경우 크기가 고정되었기 때문에 선언시에 설정한 크기보다 많은 갯수의 data가 추가되면 저장할 수 없지만, 이에 비해 Dynamic Array는 유동적이라서, data를 계속 추가하다가 기존에 할당된 메모리를 초과하게 되면, 유동적으로 size를 조절하여 데이터를 저장하는 자료구조입니다. 따라서 Dynamic Array는 size를 미리 고민할 필요없다는 장점이 있습니다.

1. doubling

▶resize를 하는 대표 방법으로 doubling이 있는데, 기존 배열의 size보다 두 배 더 큰 배열을 선언하고, 데이터를 일일이 옮기는 방법입니다. 이때 n개의 데이터를 일일이 옮겨야 하므로 시간 복잡도는 Big O n (O(n))입니다. (크기가 2배인 새로운 배열 생성 - 데이터 복사 - 참조 변수에 새 배열을 지정. 기존 배열은 Garbage Collector가 수거해감).

2. 데이터 추가(append)할 때의 시간복잡도 amortized O(1)

▶Dynamic array에 데이터를 추가할 때마다 O(1)의 시간이 걸리며, 추가를 하다가 미리 선언된 size를 넘어서는 순간에 resize를 하게 되는데, 이 때는 일일이 데이터를 모두 옮겨야 되기 때문에 O(n)의 시간이 걸리게 됩니다. 결과적으로 append의 시간복잡도는 amortized O(1)입니다. 이유는, append의 총 과정을 살펴보면 데이터를 마지막 인덱스에 추가하는 O(1)작업이 대다수이고, resize 과정 O(n)은 아주 가끔 발생합니다. 때문에 가끔 발생하는 O(n)의 resize하는 시간을, 자주 발생하는 O(1)의 작업들이 분담해서 나눠 가짐으로써 전체적으로 O(1)의 시간이 걸립니다.

● Dynamic Array를 Linked list와 비교하여 장단점을 설명해 주세요.

▶ Linked list와 비교했을때, Dynamic Array의 장점은 접근이 O(1)으로 굉장히 빠르다는 것입니다. index 접근 방식이 산술적인 연산, [배열 첫 data의 주소값] + 데이터 크기 x index로 이루어져 있기 때문입니다 (random access). 또한 순차적인 추가와 삭제, 즉, Dynamic Array 맨 뒤에 데이터를 추가/삭제하는 것이 O(1)으로 상대적으로 빠르다는 것입니다. 단점은 비순차적인 추가/삭제, 맨 끝이 아닌 곳에 data를 추가/삭제할 때 느리다는 것인데 (O(n)), 이는 Array는 메모리상에서 연속적으로 데이터들이 저장되어 있기 때문에 데이터 추가/삭제 시 뒤에 있는 data들을 모두 한 칸씩 옮겨 와야 하기 때문입니다. 또한 resize을 할 때 현저하게 낮은 performance가 발생하고, resize에 시간이 많이 걸리기 때문에 필요 이상의 memory공간을 할당 받으므로 메모리 공간 낭비가 발생한다는 단점이 있습니다.

Resources: inflearn 개발남 노씨