



실전 알고리즘 0x03강 배열

BaaaaaaaaaaaaaaaaarkingDog

목차



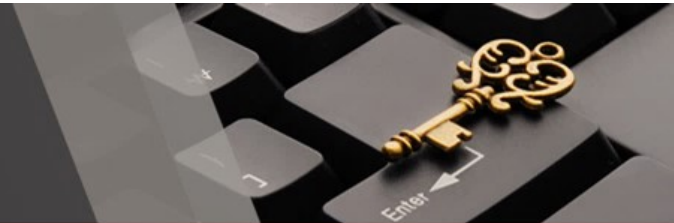
0x00 정의와 성질

0x01 기능과 구현

0x02 STL vector

0x03 연습 문제

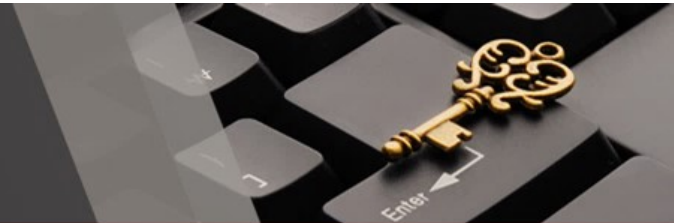
0x00 정의와 성질



0	1	2	3	4	5	6	7	8	9
2	4	6	13	-2	1	1	0	5	6

배열 - 메모리 상에 원소를 연속하게 배치한 자료구조

0x00 정의와 성질



배열의 성질

1. $O(1)$ 에 k 번째 원소를 확인/변경 가능
2. 추가적으로 소모되는 메모리의 양(=overhead)가 거의 없음
3. Cache hit rate가 높음
4. 메모리 상에 연속한 구간을 잡아야 해서 할당에 제약이 걸림

0x01 기능과 구현

임의의 위치에 있는 원소를 확인/변경, $O(1)$

0	1	2	3	4	5	6	7	8	9
2	4	6	13	-2	1	3	0	5	6



0	1	2	3	4	5	6	7	8	9
2	4	6	13	-2	1	8	0	5	6

0x01 기능과 구현

원소를 끝에 추가, $O(1)$

0	1	2	3	4	5	6	7	8	9
2	4	6	13	-2	1	3	0	5	6



0	1	2	3	4	5	6	7	8	9	10
2	4	6	13	-2	1	3	0	5	6	8

0x01 기능과 구현

마지막 원소를 제거, $O(1)$

0	1	2	3	4	5	6	7	8	9
2	4	6	13	-2	1	3	0	5	6



0	1	2	3	4	5	6	7	8
2	4	6	13	-2	1	3	0	5



0x01 기능과 구현

임의의 위치에 원소를 추가, $O(N)$

0	1	2	3	4	5	6	7	8	9
2	4	6	13	-2	1	3	0	5	6



0	1	2	3	4	5	6	7	8	9	10
2	4	6	13	-2	15	1	3	0	5	6

0x01 기능과 구현

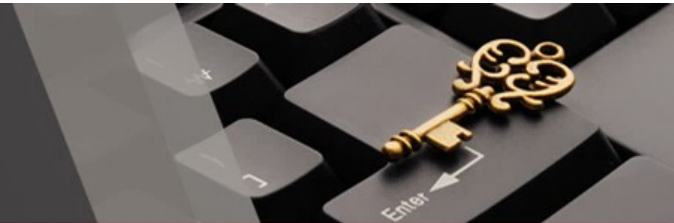
임의의 위치에 있는 원소를 제거, $O(N)$

0	1	2	3	4	5	6	7	8	9
2	4	6	13	-2	1	3	0	5	6



0	1	2	3	4	5	6	7	8
2	4	13	-2	1	3	0	5	6

0x01 기능과 구현



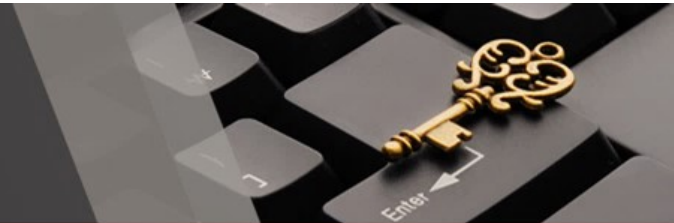
임의의 위치에 있는 원소를 확인/변경 = $O(1)$

원소를 끝에 추가 = $O(1)$

마지막 원소를 제거 = $O(1)$

임의의 위치에 원소를 추가/임의 위치의 원소 제거 = $O(N)$

0x01 기능과 구현



```
01 void insert(int idx, int num, int arr[], int& len);  
02  
03 void erase(int idx, int arr[], int& len);  
04  
05 int main(void) {  
06     int arr[10] = {10, 50, 40, 30, 70, 20};  
07     int len = 6;  
08     insert(3, 60, arr, len); // 10 50 40 60 30 70 20  
09     erase(4, arr, len); // 10 50 40 60 70 20  
10 }
```

0x01 기능과 구현

구현

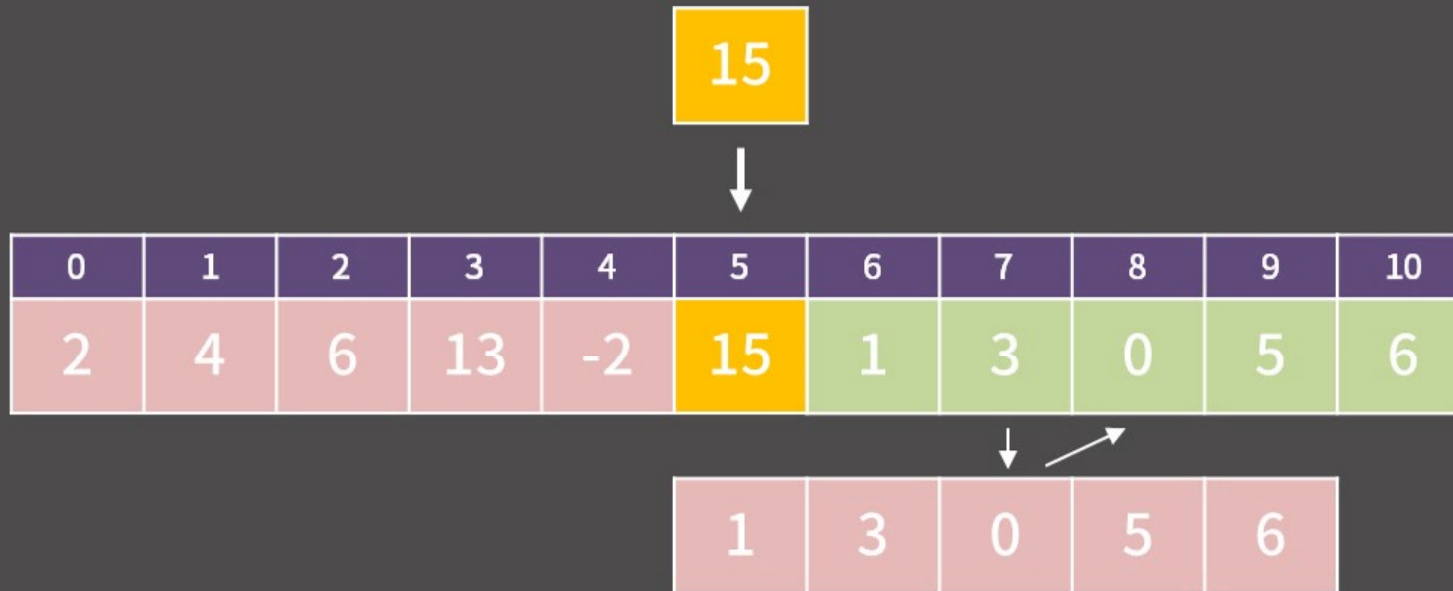
https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x03/array_test.cpp

```
01  #include <bits/stdc++.h>
02  using namespace std;
03
04  void insert(int idx, int num, int arr[], int& len){
05
06  }
07
08  void erase(int idx, int arr[], int& len){
09
10  }
11
12  void printArr(int arr[], int& len){
13      for(int i = 0; i < len; i++) cout << arr[i] << ' ';
14      cout << "\n\n";
15  }
```

```
16  void insert_test(){
17      ...
18  }
19
20  void erase_test(){
21      ...
22  }
23
24  int main(void) {
25      insert_test();
26      erase_test();
27  }
```

0x01 기능과 구현

구현 - insert 함수



0x01 기능과 구현

구현 - insert 함수

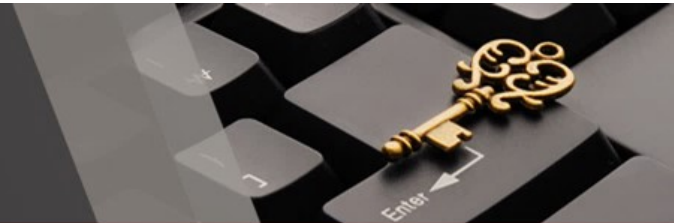
0	1	2	3	4	5	6	7	8	9	10
2	4	6	13	-2	1	1	0	5	6	-

↑
???

0	1	2	3	4	5	6	7	8	9	10
2	4	6	13	-2	15	1	3	0	5	6

↑ ↑ ↑ ↑ ↑

0x01 기능과 구현



구현 - insert 함수

```
01 void insert(int idx, int num, int arr[], int& len){  
02     for(int i = len; i > idx; i--)  
03         arr[i] = arr[i-1];  
04     arr[idx] = num;  
05     len++;  
06 }
```


0x01 기능과 구현

구현 - erase 함수

0	1	2	3	4	5	6	7	8
2	4	13	-2	1	3	0	5	6

13	-2	1	3	0	5	6
----	----	---	---	---	---	---

0x01 기능과 구현

구현 - erase 함수

0	1	2	3	4	5	6	7	8
2	4	13	-2	1	3	0	5	6

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

```
01 void erase(int idx, int arr[], int& len){  
02     len--;  
03     for(int i = idx; i < len; i++)  
04         arr[i] = arr[i+1];  
05 }
```

https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x03/array_test_ans.cpp

0x01 기능과 구현

사용 팁

```
01  int a[21];
02  int b[21][21];
03
04  // 1. memset
05  memset(a, 0, sizeof a);
06  memset(b, 0, sizeof b);
07
08  // 2. for
09  for(int i = 0; i < 21; i++)
10      a[i] = 0;
11  for(int i = 0; i < 21; i++)
12      for(int j = 0; j < 21; j++)
13          b[i][j] = 0;
14
15  // 3. fill
16  fill(a, a+21, 0);
17  for(int i = 0; i < 21; i++)
18      fill(b[i], b[i]+21, 0);
```

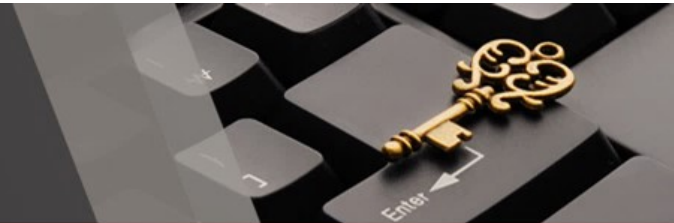
0x02 STL vector

reference : <http://www.cplusplus.com/reference/vector/vector/>

https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x03/vector_example.cpp

```
01  #include <bits/stdc++.h>
02  using namespace std;
03
04  int main(void) {
05      vector<int> v1(3, 5); // {5,5,5};
06      cout << v1.size() << '\n'; // 3
07      v1.push_back(7); // {5,5,5,7};
08
09      vector<int> v2(2); // {0,0};
10      v2.insert(v2.begin()+1, 3); // {0,3,0};
11
12      vector<int> v3 = {1,2,3,4}; // {1,2,3,4}
13      v3.erase(v3.begin()+2); // {1,2,4};
14
15      vector<int> v4; // {}
16      v4 = v3; // {1,2,4}
17      cout << v4[0] << v4[1] << v4[2] << '\n'; // 124
18      v4.pop_back(); // {1,2}
19      v4.clear(); // {}
20  }
```

0x02 STL vector



```
01  vector<int> v1 = {1,2,3,4,5,6};
02
03  // 1. range-based for loop (since C++11)
04  for(int e : v1)
05      cout << e << ' ';
06
07  // 2. not bad
08  for(int i = 0; i < v1.size(); i++)
09      cout << v1[i] << ' ';
10
11  // 3. ***WRONG***
12  for(int i = 0; i <= v1.size()-1; i++)
13      cout << v1[i] << ' ';
```

0x03 연습문제

BOJ 10808번: 알파벳 갯수

https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x03/10808_1.cpp

```
01  #include <bits/stdc++.h>
02  using namespace std;
03
04  int main(void) {
05      ios::sync_with_stdio(0);
06      cin.tie(0);
07      string s;
08      cin >> s;
09      for(char a = 'a'; a <= 'z'; a++){
10          int cnt = 0;
11          for(auto c : s)
12              if(a == c) cnt++;
13          cout << cnt << ' ';
14      }
15  }
```

0x03 연습문제

BOJ 10808번: 알파벳 갯수

a b c a z

+1
+1

a

+1

b

+1

c

d

e

•

•

•

+1

z

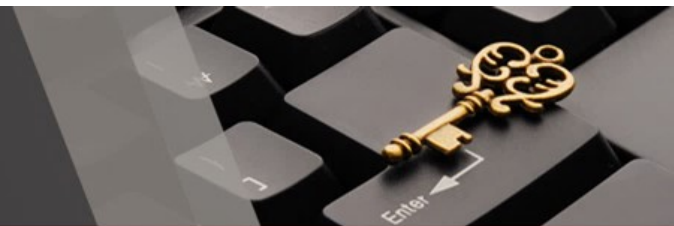
0x03 연습문제

BOJ 10808번: 알파벳 갯수

https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x03/10808_2.cpp

```
01  #include <bits/stdc++.h>
02  using namespace std;
03
04  int freq[26];
05  int main(void) {
06      ios::sync_with_stdio(0);
07      cin.tie(0);
08      string s;
09      cin >> s;
10      for(auto c : s)
11          freq[c-'a']++;
12      for(int i = 0; i < 26; i++)
13          cout << freq[i] << ' ';
14  }
```

0x03 연습문제



0x01강의 연습문제

문제 2

주어진 길이 N의 int 배열 arr에서 합이 100인 서로 다른 위치의 두 원소가 존재하면 1을, 존재하지 않으면 0을 반환하는 함수 `func2(int arr[], int N)`을 작성하라.
arr의 각 수는 0 이상 100 이하이고 N은 1000 이하이다.

`func2({1, 52, 48}, 3) = 1,`
`func2({50, 42}, 2) = 0,`
`func2({4, 13, 63, 87}, 4) = 1`

0x03 연습문제

0x01강의 연습문제

1 23 53 77 60

0x03 연습문제

0x01강의 연습문제

★ ★
1 23 53 77 60

1	23	40	47	53	60	77	99
1	1	0	0	1	0	0	0

0x03 연습문제

0x01강의 연습문제

https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x03/func2_ans.cpp

```
01 int func2(int arr[], int N){
02     int occur[101] = {};
03     for(int i = 0; i < N; i++){
04         if(occur[100-arr[i]] == 1)
05             return 1;
06         occur[arr[i]] = 1;
07     }
08     return 0;
09 }
```


강의 정리

