

[Algorithm/알고리즘](#)

[BOJ](Python) 백준 1920번: 수 찾기

그리브 | 2023. 11. 14. 01:48 | 수정 | 삭제

문제

문제
풀이<https://www.acmicpc.net/problem/1920>



1920번: 수 찾기

첫째 줄에 자연수 N ($1 \leq N \leq 100,000$)이 주어진다. 다음 줄에는 N 개의 정수 $A[1], A[2], \dots, A[N]$ 이 주어진다. 다음 줄에는 M ($1 \leq M \leq 100,000$)이 주어진다. 다음 줄에는 M 개의 수들이 주어지는데, ...

www.acmicpc.net

주제	해시 테이블
시간 / 메모리 제한	1초 / 128MB
정답 비율	29.789%

N 개의 정수 $A[1], A[2], \dots, A[N]$ 이 주어져 있을 때,

이 안에 X 라는 정수가 존재하는지 알아내는 프로그램을 작성하시오.

입력

첫째 줄에 자연수 N ($1 \leq N \leq 100,000$)이 주어진다.

다음 줄에는 N 개의 정수 $A[1], A[2], \dots, A[N]$ 이 주어진다.

다음 줄에는 M ($1 \leq M \leq 100,000$)이 주어진다.

다음 줄에는 M 개의 수들이 주어지는데, 이 수들이 A 안에 존재하는지 알아내면 된다.

- 모든 정수의 범위는 $(-2)^{31}$ 보다 크거나 같고 2^{31} 보다 작다.

출력

M 개의 줄에 답을 출력한다.

존재하면 1을, 존재하지 않으면 0을 출력한다.

- 예제 입력 1

Copy

```
1 5
2 4 1 5 2 3
3 5
```

```
4 1 3 7 9 5
```

python

- 예제 출력 1

```
1 1
2 1
3 0
4 0
5 1
```

Copy

python

풀이

위 문제를 보고 바로 작성할 수 있는 코드를 적어 실행해 봤더니 역시나

아니나 다를까 시간초과가 발생하였다.

```
1 import sys
2
3 N = int(sys.stdin.readline())
4 given = list(map(int, sys.stdin.readline().split()))
5
6 M = int(sys.stdin.readline())
7 numbers = list(map(int, sys.stdin.readline().split()))
8
9 for num in numbers:
10     if num in given:
11         print('1')
12     else:
13         print('0')
```

Copy

python

위의 경우 numbers를 순회하며 $O(N)$ 의 연산을 수행하고

in 연산에서 다시 $O(N)$ 연산을 수행하기 때문에

총 $O(N^2)$ 의 연산이 사용되었고 이 경우 시간초과가 발생한다는 것을 알 수 있다.

1) 이진 탐색

저번 백준 문제 풀이에 사용한 이진 탐색 트리는 $O(N \log N)$ 연산이므로

이를 사용한 코드를 작성해보았다.

```
1 import sys
2
3 def binary_search(list, target, start, end):
4     if start > end:
5         return 0
6     mid = (start + end) // 2
7     if list[mid] == target:
8         return 1
9     elif list[mid] > target:
10        return binary_search(list, target, start, mid - 1)
11    else:
12        return binary_search(list, target, mid + 1, end)
13
14 N = int(sys.stdin.readline())
```

Copy

```

15 given = list(map(int, sys.stdin.readline().split()))
16
17 sorted_list = sorted(given)
18
19 M = int(sys.stdin.readline())
20 numbers = list(map(int, sys.stdin.readline().split()))
21
22 for num in numbers:
23     print(binary_search(sorted_list, num, 0, len(sorted_list) - 1))

```

python

위와 같이 이진 탐색을 이용하여 풀이하니 시간 초과가 발생하지 않았다.

문제	결과	메모리	시간	언어	코드 길이
1920	맞았습니다!!	48200 KB	504 ms	Python 3 / 수정	611 B

2) 해시테이블

이진 탐색을 이용한 풀이로 이미 정답은 맞추었지만

해시테이블을 이용한 풀이도 가능하다.

파이썬에서는 딕셔너리와 set 자료형을 이용하면 해시테이블을 사용할 수 있는데,

딕셔너리가 더욱 익숙하므로 딕셔너리를 이용해 해시테이블 코드를 작성해보았다.

딕셔너리는 해시 구조로 이루어져 있기 때문에

바로 해시테이블 알고리즘 풀이가 가능하다.

```

1 import sys
2
3 N = int(sys.stdin.readline())
4 given = list(map(int, sys.stdin.readline().split()))
5 given_dict = {}
6 for key in given:
7     given_dict[key] = ' '
8 M = int(sys.stdin.readline())
9 numbers = list(map(int, sys.stdin.readline().split()))
10
11 for num in numbers:
12     if num in given_dict:
13         print('1')
14     else:
15         print('0')

```

Copy

python

위와 같이 코드를 작성해주었다.

딕셔너리의 value 영역은 낭비하고 key 영역만 사용하여 다소 비효율적이며 보일 수 있지만,

실제로 실행해 보았을 때 아래와 같은 결과를 확인할 수 있었다.

문제	결과	메모리	시간	언어	코드 길이
1920	맞았습니다!!	52352 KB	172 ms	Python 3 / 수정	339 B

이진탐색을 이용한 풀이와 연산에 소요된 시간을 비교했을 때

약 3배 이상 빨리 수행된 것을 확인할 수 있다.

딕셔너리, 즉 해시 테이블 연산의 효율성을 눈으로 확인하였다.



♡ 공감 📌 📄 000

'Algorithm' > 알고리즘' 카테고리의 다른 글

[BOJ](Python) 백준 1620번: 나는야 포켓몬 마스터 이다솜 (0)	2023.11.07
[휴학 중 코테 부수기] # 5. 정렬 알고리즘 (0)	2023.11.06
[휴학 중 코테 부수기] # 4. 그래프 탐색 알고리즘: DFS/BFS (2)	2023.11.06
[휴학 중 코테 부수기] # 3. 구현(implementation) (1)	2023.11.04
[휴학 중 코테 부수기] # 2. 그리디 알고리즘 (6)	2023.10.23

'Algorithm/알고리즘' Related Articles

	메모리	NO IMAGE		
	57692 KB			

[BOJ](Python) 백준 1620
번: 나는야 포켓몬 마스터 이...

[휴학 중 코테 부수기] # 5. 정
렬 알고리즘

[휴학 중 코테 부수기] # 4. 그
래프 탐색 알고리즘:...

[휴학 중 코테 부수기] # 3. 구
현(Implementation)

☐ Secret

안녕하세요! 어떤 댓글이든 환영합니다! 

댓글달기