

알고리즘

해시, 이진검색, 힙

이영석

학습내용

- 자료구조 리뷰1
 - 해시, 힙
- 자료구조 교과목에서 학습 내용
 - 해시, 힙의 이론과 직접 구현
 - 언어별(Java, Python, C/C++) STL 해시, 힙 사용 방법
- 알고리즘 수업
 - STL 활용 응용문제

문제 1

완주하지 못한 선수

문제 설명

수많은 마라톤 선수들이 마라톤에 참여하였습니다. 단 한 명의 선수를 제외하고는 모든 선수가 마라톤을 완주하였습니다.

마라톤에 참여한 선수들의 이름이 담긴 배열 `participant`와 완주한 선수들의 이름이 담긴 배열 `completion`이 주어질 때, 완주하지 못한 선수의 이름을 `return` 하도록 `solution` 함수를 작성해주세요.

제한사항

- 마라톤 경기에 참여한 선수의 수는 1명 이상 100,000명 이하입니다.
- `completion`의 길이는 `participant`의 길이보다 1 작습니다.
- 참가자의 이름은 1개 이상 20개 이하의 알파벳 소문자로 이루어져 있습니다.
- 참가자 중에는 동명이인이 있을 수 있습니다.

입출력 예

participant	completion	return
["leo", "kiki", "eden"]	["eden", "kiki"]	"leo"
["marina", "josipa", "nikola", "vinko", "filipa"]	["josipa", "filipa", "marina", "nikola"]	"vinko"
["mislav", "stanko", "mislav", "ana"]	["stanko", "ana", "mislav"]	"mislav"

입출력 예 설명

예제 #1

"leo"는 참여자 명단에는 있지만, 완주자 명단에는 없기 때문에 완주하지 못했습니다.

예제 #2

"vinko"는 참여자 명단에는 있지만, 완주자 명단에는 없기 때문에 완주하지 못했습니다.

예제 #3

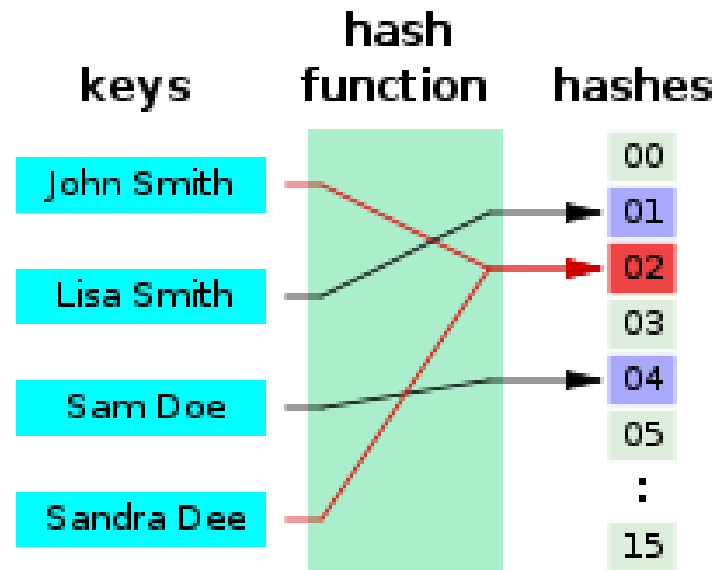
"mislav"는 참여자 명단에는 두 명이 있지만, 완주자 명단에는 한 명밖에 없기 때문에 한명은 완주하지 못했습니다.

해결방법

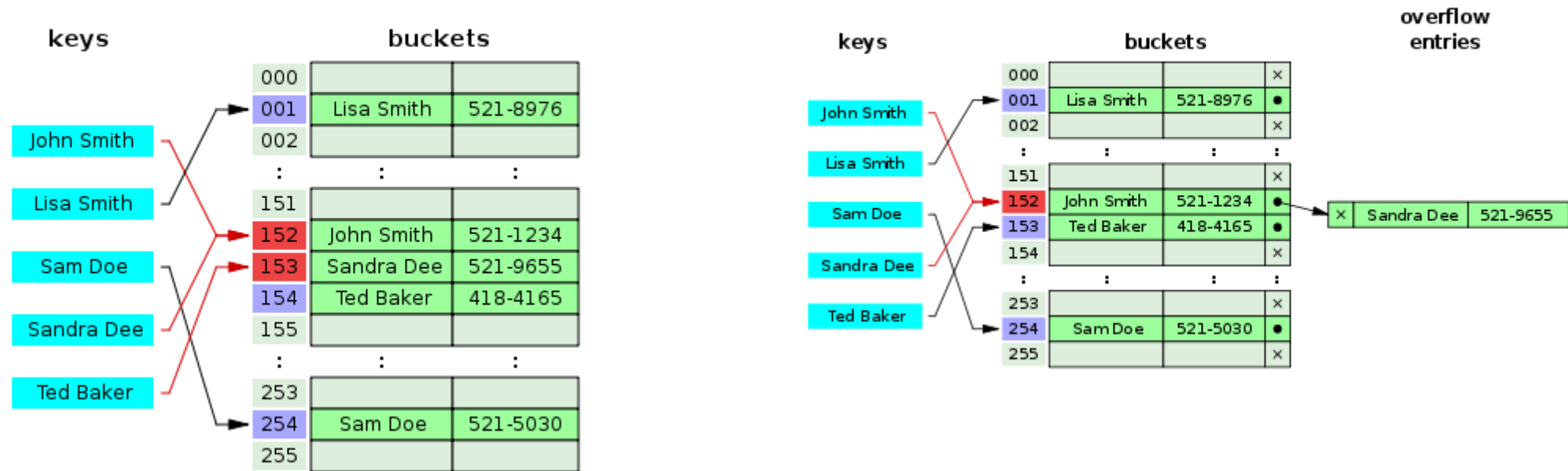
- 두 개의 배열을 비교?

해시

- Hash Table or Hash Map?
 - 키에 대한 해시 값을 이용하여 값을 저장, 조회 키-값 쌍의 개수 동적 변화 associate array (Map, Dictionary, Symbol Table)
 - Collision: open addressing with chaining



Hash Collision



https://en.wikipedia.org/wiki/Hash_table#Collision_resolution

해시 구현

- C++
 - unordered_set, unordered_map: hash
 - set, map: 이진검색트리
- Java
 - Collections Framework (HashMap, Hashtable은 HashMap인터페이스 구현)
 - HashMap: separate chaining (list vs tree)
- Python
 - Dictionary
 - Collections

Python modules

- collections
 - dict, list, set 및 tuple
 - namedtuple, deque, Counter, OrderedDict, defaultdict, ChainMap
- heapq

문제2

수 찾기

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	40109	10938	7120	27.917%

문제

N개의 정수 $A[1], A[2], \dots, A[N]$ 이 주어져 있을 때, 이 안에 X라는 정수가 존재하는지 알아내는 프로그램을 작성하시오.

입력

첫째 줄에 자연수 $N(1 \leq N \leq 100,000)$ 이 주어진다. 다음 줄에는 N개의 정수 $A[1], A[2], \dots, A[N]$ 이 주어진다. 다음 줄에는 $M(1 \leq M \leq 100,000)$ 이 주어진다. 다음 줄에는 M개의 수들이 주어지는데, 이 수들이 A안에 존재하는지 알아내면 된다. 모든 정수들의 범위는 int 로 한다.

출력

M개의 줄에 답을 출력한다. 존재하면 1을, 존재하지 않으면 0을 출력한다.

예제 입력 1 복사

```
5
4 1 5 2 3
5
1 3 7 9 5
```

예제 출력 1 복사

```
1
1
0
0
1
```

해결방법

- 순차검색
- 정렬 후 이진검색
 - 정렬 구현?
 - 이진탐색 구현?
- 해시
- 복잡도
 - $O(n)$, $O(n \log n)$, $O(n)$

문제3

입국심사

[dark](#)

문제 설명

n 명이 입국심사를 위해 줄을 서서 기다리고 있습니다. 각 입국심사대에 있는 심사관마다 심사하는데 걸리는 시간은 다릅니다.

처음에 모든 심사대는 비어있습니다. 한 심사대에서는 동시에 한 명만 심사를 할 수 있습니다. 가장 앞에 서 있는 사람은 비어 있는 심사대로 가서 심사를 받을 수 있습니다. 하지만 더 빨리 끝나는 심사대가 있으면 기다렸다가 그곳으로 가서 심사를 받을 수도 있습니다.

모든 사람이 심사를 받는데 걸리는 시간을 최소로 하고 싶습니다.

입국심사를 기다리는 사람 수 n , 각 심사관이 한 명을 심사하는데 걸리는 시간이 담긴 배열 `times`가 매개변수로 주어질 때, 모든 사람이 심사를 받는데 걸리는 시간의 최솟값을 `return` 하도록 `solution` 함수를 작성해주세요.

제한사항

- 입국심사를 기다리는 사람은 1명 이상 1,000,000,000명 이하입니다.
- 각 심사관이 한 명을 심사하는데 걸리는 시간은 1분 이상 1,000,000,000분 이하입니다.
- 심사관은 1명 이상 100,000명 이하입니다.

입출력 예

n	times	return
6	[7, 10]	28

입출력 예 설명

가장 첫 두 사람은 바로 심사를 받으러 갑니다.

7분이 되었을 때, 첫 번째 심사대가 비고 3번째 사람이 심사를 받습니다.

10분이 되었을 때, 두 번째 심사대가 비고 4번째 사람이 심사를 받습니다.

14분이 되었을 때, 첫 번째 심사대가 비고 5번째 사람이 심사를 받습니다.

20분이 되었을 때, 두 번째 심사대가 비지만 6번째 사람이 그곳에서 심사를 받지 않고 1분을 더 기다린 후에 첫 번째 심사대에서 심사를 받으면 28분에 모든 사람의 심사가 끝납니다.

해결 방법

- 6명이 대기
- 2명의 심사관 A, B 각각 7분, 10분
- 최악의 경우
 - 6명 모두 B 심사관에 대기: 60분
- 풀이
 - 60분보다 줄이는 경우는? 잘 모르겠다. 대충 짚어서 찾아보자. 절반?
 - 30분으로 6명 모두 심사할 수 있을까?
 - 30분이면 A는 4명, B는 2명 -> 30분안에는 6명 모두 가능
 - 30분보다 더 작게 할 수 있을까? 대충 절반? 15분?
 - 15분이면 A는 2명, B는 1명 -> X -> 다시 시간을 늘려야함! 15+30의 절반? 22분?

문제4

예산

문제 설명

국가의 역할 중 하나는 여러 지방의 예산요청을 심사하여 국가의 예산을 분배하는 것입니다. 국가예산의 총액은 미리 정해져 있어서 모든 예산요청을 배정해 주기는 어려울 수도 있습니다. 그래서 정해진 총액 이하에서 **가능한 한 최대의** 총 예산을 다음과 같은 방법으로 배정합니다.

1. 모든 요청이 배정될 수 있는 경우에는 요청한 금액을 그대로 배정합니다.
2. 모든 요청이 배정될 수 없는 경우에는 특정한 정수 상한액을 계산하여 그 이상인 예산요청에는 모두 상한액을 배정합니다. 상한액 이하의 예산요청에 대해서는 요청한 금액을 그대로 배정합니다.

예를 들어, 전체 국가예산이 485이고 4개 지방의 예산요청이 각각 120, 110, 140, 150일 때, 상한액을 127로 잡으면 위의 요청들에 대해서 각각 120, 110, 127, 127을 배정하고 그 합이 484로 가능한 최대가 됩니다.

각 지방에서 요청하는 예산이 담긴 배열 `budgets`과 총 예산 `M`이 매개변수로 주어질 때, 위의 조건을 모두 만족하는 상한액을 `return` 하도록 `solution` 함수를 작성해주세요.

제한 사항

- 지방의 수는 3 이상 100,000 이하인 자연수입니다.
- 각 지방에서 요청하는 예산은 1 이상 100,000 이하인 자연수입니다.
- 총 예산은 `지방의 수` 이상 1,000,000,000 이하인 자연수입니다.

입출력 예

<code>budgets</code>	<code>M</code>	<code>return</code>
[120, 110, 140, 150]	485	127

출처

※ 공지 - 2019년 3월 15일, 테스트케이스가 강화되었습니다.

이번 업데이트로 인해 지방의 수가 최대 10,000개에서 100,000개로 늘어났으며, 이에 따라 테스트케이스가 수정되었습니다.

이로 인해 이전에 통과하던 코드가 더 이상 통과하지 않을 수 있습니다.

이진 검색

- C++
 - STL set, map, multiset, multimap
- Java
 - Arrays.binarySearch(sorted Array, key)
 - Collections.binarySearch(sortedList, key)
- Python
 - Bisect.bisect

```
import bisect
mylist = [1, 2, 3, 7, 9, 11, 33]
print(bisect.bisect(mylist, 3))
```

문제5

문제 설명

라면 공장에서는 하루에 밀가루를 1톤씩 사용합니다. 원래 밀가루를 공급받던 공장의 고장으로 앞으로 k 일 이후에야 밀가루를 공급받을 수 있기 때문에 해외 공장에서 밀가루를 수입해야 합니다.

해외 공장에서는 향후 밀가루를 공급할 수 있는 날짜와 수량을 알려주었고, 라면 공장에서는 운송비를 줄이기 위해 최소한의 횟수로 밀가루를 공급받고 싶습니다.

현재 공장에 남아있는 밀가루 수량 `stock`, 밀가루 공급 일정(`dates`)과 해당 시점에 공급 가능한 밀가루 수량(`supplies`), 원래 공장으로부터 공급받을 수 있는 시점 k 가 주어질 때, 밀가루가 떨어지지 않고 공장을 운영하기 위해서 최소한 몇 번 해외 공장으로부터 밀가루를 공급받아야 하는지를 `return` 하도록 `solution` 함수를 완성하세요.

`dates[i]`에는 i 번째 공급 가능일이 들어있으며, `supplies[i]`에는 `dates[i]` 날짜에 공급 가능한 밀가루 수량이 들어 있습니다.

제한사항

- `stock`에 있는 밀가루는 오늘(0일 이후)부터 사용됩니다.
- `stock`과 k 는 2 이상 100,000 이하입니다.
- `dates`의 각 원소는 1 이상 k 이하입니다.
- `supplies`의 각 원소는 1 이상 1,000 이하입니다.
- `dates`와 `supplies`의 길이는 1 이상 20,000 이하입니다.
- k 일 때에는 밀가루가 충분히 공급되기 때문에 $k-1$ 일에 사용할 수량까지만 확보하면 됩니다.
- `dates`에 들어있는 날짜는 오름차순 정렬되어 있습니다.
- `dates`에 들어있는 날짜에 공급되는 밀가루는 작업 시작 전 새벽에 공급되는 것을 기준으로 합니다. 예를 들어 9일째에 밀가루가 바닥나더라도, 10일째에 공급받으면 10일째에는 공장을 운영할 수 있습니다.
- 밀가루가 바닥나는 경우는 주어지지 않습니다.

입출력 예

stock	dates	supplies	k	result
4	[4,10,15]	[20,5,10]	30	2

입출력 예 설명

- 현재 밀가루가 4톤 남아 있기 때문에 오늘과 1일 후~3일 후까지 사용하고 나면 모든 밀가루를 다 사용합니다. 따라서 4일 후에는 반드시 밀가루를 공급받아야 합니다.
- 4일째 공급받고 나면 15일 이후 아침에는 9톤의 밀가루가 남아있게 되고, 이때 10톤을 더 공급받으면 19톤이 남아있게 됩니다. 15일 이후부터 29일 이후까지 필요한 밀가루는 15톤이므로 더 이상의 공급은 필요 없습니다.
- 따라서 총 2회의 밀가루를 공급받으면 됩니다.

해결 방법

- 현재 재고 stock이 없어지기전 수입 가능 데이터를 기억하기
 - 수입할 공급량 max heap 저장
- 재고 stock이 k 보다 작으면 수입!
 - Max heap의 최대값

Hash, Binary Search, Heap

- 언어별 사용법
- 대표 문제