

알고리즘 응용

- Lab 09. binary search -

제출일자	2021.05.09
분 반	00
이 름	강인한
학 번	201701969

시간을 이진 탐색 방식을 이용하여 걸린 시간을 찾았다. 먼저 시간이 주어질 때 가장 많이 계산할 수 있는 경우는 주어진 시간을 각각의 계산대가 소요하는 시간을 나눈 값을 다 합치면 구할 수 있다. 즉 10초가 주어졌을 때 2초의 계산대와 3초의 계산대가 존재한다면 총 $10//2 + 10//3$ 인 8이 된다. 시간을 특정 숫자로 가정했을 때 가장 많이 계산할 수 있는 수를 구할 수 있으므로 반대로 생각해 시간을 찾아가며 ex) 7명의 계산을 할 수 있는 가장 적은 시간을 찾는다. 찾는 방법은 앞에서 말했듯이 이진 탐색을 이용한다. 먼저 계산에 걸리는 시간을 줄이기 위해 시간의 최솟값을 예상해보면 모든 계산대가 걸리는 시간이 같은 경우이다. 그렇게 되면 줄 서 있는 사람의 수가 계산대의 수보다 적은 경우, 많은 경우로 나뉘어서 생각해 야겠지만 결론적으로는 계산에 걸리는 시간(모두 동일) 만큼의 최솟값을 갖게 된다. 최댓값으로는 계산대 중에서 가장 적게 시간이 걸리는 계산대가 모든 계산을 다 하게 되었을 경우가 된다. 즉 $\min(\text{time}) * k$ 가 된다(k =고객의 수). 그럼 최솟값을 left로 하고 최댓값을 right로 하여 이진 탐색을 시작한다. left와 right의 중앙값을 계산에 걸리는 시간으로 각각 나눠 더해 주는 작업을 하여 원하는 고객을 받을 수 있는 최소한의 시간을 구한다.

```
while True:
    mid = (left+right) // 2
    sumMidDivTime = 0
    for i in range(n):
        sumMidDivTime += mid//time[i]
    if sumMidDivTime < k:
        left = mid + 1
    else:
        result = mid
        right = mid-1

    if left > right:
        break
```

```
C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/use
6 7
19 12 2 7 6 16
10

Process finished with exit code 0
```

이진 탐색은 $\log n$ 의 시간복잡도를 갖는다. 이 코드에서 k 라는 고객의 수가 있을 때의 시간 복잡도는 $left$ 는 $\min(time)$ 으로 정하고 $right$ 는 $\min(time)*k$ 로 정했으므로 구하고자 하는 시간은 $\log (\min(time)*k - \min(time))$ 이 되고 while문 내에서 for문을 이용해 k 만큼의 반복을 하였으므로 빅오 표기법 시간복잡도는 $k(\log k) n$ 으로 표시해보면 $n \log n$ 이 된다.