매일매일 백준 실수 및

깨달음 노트

By. 류이안

2/11(토) ~ 2/15(수): 노트를 쓰지 않았음

2/11 (토): 14620번, 1912번 해결

2/12 (일): 2309, 2231, 3085, 10448번 해결

2/13 (월): 일정

2/14 (화): 휴식

2/15 (수): 1018번, 1182번 해결

2/16 (목)

* 4796번
  + Print 함수에서 + 로 합쳐진 string을 출력할 때, int형 변수는 모두 string형으로 바꿔주자
  + divmod에서 첫 번째 원소가 몫이다. # index = 0
  + result에 divmod(camp[2], camp[1])[1]을 항상 더하는 것으로 생각했음 => 1 20 39 인 경우 반례 => camp[0]과 비교해 작은 것을 더해주자
* 1449번: 실수 없음

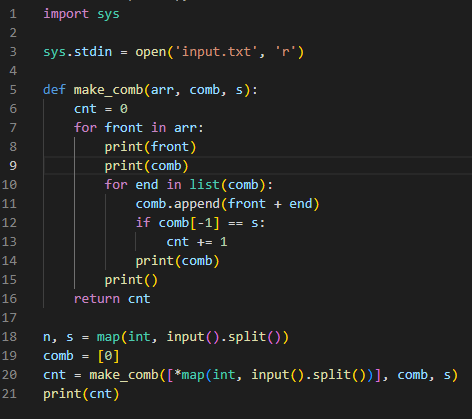
2/17 (금)

* 17509번: 실수 없음
* 11047번
  + 궁금증: “와 ‘의 차이 – 문법적으로는 차이 없음. 사람 따라 구분해서 씀
  + 궁금증: list()와 []의 차이 – map 함수 밖에 쓸 때 list는 그걸 그냥 list로 만들어 줌
  + 역으로 정렬하는 함수를 까먹었음 – sort(reverse = True)
  + 끝나는 시간을 기준으로만 sort하면 앞 시간이 언제든 상관없이 sort가 되어버리므로, 앞 시간 기준으로 먼저 sort를 한 번 해주자
* 11000번
  + 이 문제는 11047번 문제와 다르게 시간 내 최대 가능 개수를 찾는게 아니라서 앞 번호를 기준으로 sort를 해주는게 옳음
  + 풀다가 자러감

2/18 (토)

* 11000번
  + 해결 완료
* 1700번
  + Input 뒤에 () 붙이는거 까먹지 않기
  + Value error가 떴다 – remove에 temp가 0으로 들어가는듯? – 이유는 계속 꼽아 놓을 플러그를 가리키는 temp\_plugs에 길이가 확인을 먼저 안하고 먼저 무지성으로 넣고 나서 길이를 따진 것이 문제였다.
  + Value error가 한 번 더 떴다 – 길이가 총 플러그보다 작을 때 무지성으로 이미 존재하는 숫자인지 확인 안하고 넣던 것이 문제였다.
  + 이번엔 정답이 틀렸다 – temp\_plugs에 값을 넣을 때 이번에도 중복 확인을 안했다. => 중복 미확인으로 실수 두개가 나왔다. 주의

2/19 (일)

* 
* 더 나은 시간 복잡도의 1182번 코드를 공부했다.
* 2212번
  + 문제 자체를 처음에 잘 못 이해했었다. 각 센서들과 비용을 총 계산 하는게 아니라 하나의 기지국이 갖는 범위가 코스트가 되는 것.
  + for문에 배열 길이 넣을 때 len을 자꾸 까먹음
  + 인덱스 개수가 0개인지 아닌지 항상 고려하자

2/20 (월)

* 13904번
  + Input 받을 때 자료형 항상 고려하는거 까먹지 않기
  + 이 문제는 처음에 주는 수가 과제 개수이므로 이걸 최대 기간 제한으로 두는 실수를 하면 안됐음
* 15748번: 실수 없음
  + 백준에서 돌릴 때 input = sys.stdin.readline을 사용하면 훨씬 빠르다는 것을 알았다. 신기하다
* 1493번
  + 풀다가 자러감

2/21 (화)

* 1493번: 실수 없음

2/22 (수) ~ 2/26 (일): 일정이 있어 풀지 못함

2/27 (월)

* 1629번
  + 잊고 있던 Module의 성질을 기억하고 난 후 해결 되었다. Divide & Conquer을 단순하게 사용하는 것 만으로는 풀리지 않았고, mod를 단계마다 계속 취해주어도 나머지는 변하지 않는다는 것을 기억해야 한다.
  + Pow 함수를 사용했을 때 실행 시간이 간단하면서도 매우 짧게 나온다!
* 2104번
  + 풀다 피곤해서 자러감

2/28 (화)

* 2104번
  + 매번 sum이나 min 함수를 써주는 것에는 시간이 꽤 든다는 것을 잊지 말자
  + 배열 index 다루기를 항상 신경 써서 해주자

3/1 (수)

* 1725번: 실수 없음

3/2 (목): 푼 문제 없음

3/3 (금)

* 1780번
  + 전부 같은 숫자인지 확인 할 때, 나머지는 하나도 없는지도 확인해야 된다.

3/4 (토)

* 1992번, 1074번: 실수 없음

3/5 (일)

* 2447번
  + For문 안에서 for문을 쓸 때 이미 쓴 문자 다시 안 쓰도록 조심
* 2339번: 문제 자체에 헷갈린 것이 많음.

3/6 ~ 3/17은 풀지 못한 날도 있고 기록을 빼먹었음

3/18 (토)

* 12865번
  + 이번 문제는 이전 문제와 다르게 한 물건을 여러 개 넣을 수 없다는 것을 놓쳤다.
  + => 이 경우 물건 탐색을 먼저해야함, 그리고 무게도 큰 무게부터 탐색해야함
* 16500번
  + 왠진 모르겠지만 sys.stdin.readline이 없어야 통과가 된다

3/19 (일)

* 16500번
  + sys.stdin.readline을 했을 때 이번엔 string형 입력을 다루므로 \n이 같이 들어와 버려서 문제가 된다.
* 11055번
  + 그 값이 최대 수열일 경우를 확인하지 않음

3/20 (월)

* 1351, 1354번
  + Dictionary를 쓰면 간단한 문제인데, 생각을 전혀 안 하고 있었다. 다양한 생각을 하는 연습을 하자.
  + 별해로는 일정 수까지는 바텀업으로 풀고 그 이상은 탑다운으로 푸는 방식이 있다. 수 기준은 계산으로 정하면 된다.
* 2437번
  + 생각의 전환을 빠르게 하도록 하자

3/21 (화)

* 2805, 2512번
  + 문제 없음
* 2343, 6236번
  + 최소 값을 설정할 때 무조건 0으로 설정을 해버리면 안 될 경우가 있다는 것을 잊지 말자.
* 1654번
  + 최대 값을 선 길이 중 최대로 하거나 할 때 그거 보다 1 크거나 작은 수로 잡아주도록 하자. 그래야 최대 길이일 때를 무시하지 않는다.
* 2110번: 실수 없음

3/22 (수)

* 16434번
  + HP가 0이 되자마자 break를 걸어주는 것을 안 함. 적의 HP를 정확히 0으로 만들었을 때 HP를 소모 안하는 것을 고려 안 함.
* 15732번
  + 도토리 박스들 사이에 값이 결과값으로 나오는 경우를 주의하자.
* 1300번
  + 아이디어를 좀 더 정밀하게 짜자

3/23 (목): 휴식

3/24 (금)

* 1021번: 실수 없음

3/25 (토)

* 1158번, 2346번: 실수 없음
* 2840번
  + 문제 조건을 항상 꼼꼼하게 읽도록 하자

3/26 (일)

* 1406번
  + 나는 linked list class를 새로 만들어서 풀었다. 그런데, 커서 왼쪽 문자들과 오른쪽 문자들을 서로 분리해서 각각 리스트에 저장하는 방식으로 풀면 더 빠르게 풀 수 있다.
* 5397번
  + List에 문자열이 들어가 있을 때, 출력을 할 때는 print(“”.join(~~)) 와 같은 식으로 출력 하는 것이 for문을 써서 전부 도는 것보다 훨씬 빠르다.
  + If 조건문에서 list의 길이가 0인지 확인 할 때는 그냥 그 list를 조건으로 집어넣어도 확인이 가능하다.
  + Del [-1]보다는 pop()이 더 빠르다.
  + 역순으로 슬라이싱을 할 때는, [::-1]을 사용하면 된다.

3/27 (월)

* 3190번
  + 간 경로를 모두 기록해서 체크하는 방식을 사용했지만, 현재 위치 정보를 판에 기록하는 방식을 사용하면 훨씬 빨리 풀 수 있다.
* 1918번
  + 여러가지 케이스를 다 고려하지 않았다.

3/28 (화)

* 1935번: 실수 없음
* 1725번
  + 블로그 글을 보고 divide and conquer를 썼었던 문제를 스택을 이용해 O(N)으로 재 구현 해보았다. 짜다가 중간에 이해를 잘못한 부분들이 있어서 꼬였는데, 이 때는 그냥 코드를 엎고 다시 짜자. 괜히 기능 조금씩 바꾸려다 헷갈려서 더 오래 걸렸다.
* 10828, 9012번: 실수 없음

3/29 (수)

* 2304번: 실수 없음
* 2841, 3986번: 문제 조건 하나 헷갈림
* 5076번
  + Pop()을 할 때 항상 list가 비어 있지는 않은지 확인해주도록 하자.
  + 리스트가 비어 있는지 여부를 확인하여 결과를 내 놓을 때, 틀린 결과인데도 리스트를 비게 내놓는 경우는 없는지 확인하자. 31번째 줄에서 원래는 빈 리스트를 내놓았었다.

3/30 (목)

* 10845, 2164, 1966번: 실수 없음

3/31 (금) ~ 4/1 (토): 집안 일정

4/2 (일)

* 3078, 10866번: 실수 없음
* 5430번
  + input은 무조건 다 받아야함. 빈칸이라고 안받으면 문제 생김.
  + 출력의 양식을 꼭 확인하자. list를 다룬다고 해서 list 출력 그대로 출력하면 안된다. 스페이스가 붙어있으면 붙여서 출력해야한다. 설마가 아니다. 출력 양식을 그대로 출력하도록 하자

4/3 (월) ~ 4/5 (수): 개인 사정

4/6 (목)

* 1003번: 실수 없음

4/7 (금)

* 1012번
  + ^ 연산자를 사용할 때는 양 쪽에 괄호를 활용 해 주자.
  + dfs 할 때 모든 연결 된 노드를 다 돌아봐야 한다는 것을 명심하자.
  + setrecursionerror에 너무 큰 수를 넣으면 overflow error가 뜬다.
  + 두 번째로 좌표를 배열에 일일이 찍어서 푸는 방식으로도 풀어보았다.
* 9466번
  + 푸는 방식을 여러 번 보면서 확실하게 익히도록 하자. 처음에 꽤 헷갈렸었다.
  + 이 알고리즘 외에도 덱으로 싸이클 포함 여부를 확인하는 법이 있다.

4/8 (토)

* 11724번: 실수 없음

4/9 (일)

* 1743번: 실수 없음

4/10 (월)

* 2667번
  + String을 받을 때 개행문자 strip 하는 것을 또 까먹었다. 항상 잊지 말자.

4/11 (화)

* 2583번
  + 출력 조건을 제대로 인지하지 않고 제출했다. 실수 하지 말자.
* 10026번
  + 처음에 생각난 방법인 G를 R로 치환해서 구하는 방법으로 했을 때는 바로 맞았고, 그 다음으로 생각난 방법은 계속 틀리는 중.
  + 그 다음 생각난 방법으로는 해결이 안 된다.
* 11403번
  + 피곤 했는지 dfs를 하고 있으면서 visited 리스트 활용을 못 했다.
  + 그런데 이후 visited를 1차원으로만 구성해도 된다는 것을 간과했다.
  + 1차원으로 하니까 풀렸고, 좋은 코드를 보며 아이디어를 참고하여 직접 다시 코드를 짜보았다

4/12 (수): 문제 풀지 않음

4/13 (목)

* 2468번, 10552, 15649번: 실수 없음

4/14 (금)

* 10265번
  + 처음 아이디어를 얻는데 시간이 좀 걸렸다. Cycle의 크기가 최소고, components의 최대 크기가 최대인 것을 생각하지 못했다.
  + 이후 components와 cycle을 알아내는데 고려하지 않은 부분이 있어서 실수가 좀 있었다.

4/15 (토) ~ 4/30 (일): 개인 사정과 여러 악재가 겹쳐 공부하지 못함

5/1 (월)

* 2644번: 실수 없음

5/2 (화)

* 2178번: 실수 없음

5/3 (수)

* 7562번, 5014번: 실수 없음

5/4 (목)

* 1260번: 문제 조건을 잘 읽자. 숫자 적은 순부터 방문하는 조건을 놓쳤다.

5/6 (토)

* 6593번: 출력 형식 실수 및 출구 위치를 첫 층과 마지막 층에 있을 것이라고 단정 지어 생각 했음.