**0821**

**1) 프로그래머스 문자열 압축 문제**

배운점)

* Index를 정확하게 파악하지 않고 테스트해서 경험적으로 고치는 습관!
* 문제를 정확하게 읽지 않고 푼다. 인덱스를 하나씩 추가해야할 경우는 없었는데, 있는 것으로 착각하여 예외처리를 했다.

잘한점)

* 이전에 풀었던 코드보다 한층 가독성이 좋아졌다! 아이디어를 쉽게 떠올렸다. Ex) 반복문을 통해서 1 단위로 압축, 2 단위로 압축, 3 단위로 압축하면 되겠구나.

**2) 프로그래머스 무지의 먹방 라이브**

배운점)

* 우선순위 큐를 써야하는 아이디어를 떠올리지 못했다. -> 작은 값을 배열에서 선택하고, 삭제도 해야한다면 우선순위 큐가 유리함. (딕셔너리로 구현해봤는데 매우 어려웠음. )
* 무조건 구현으로만 해결할 수 있다고 생각했다. 이 문제에서는 그리디 접근으로 푸는 아이디어 문제!

**3) 백준 피보나치 수열**

배운점)

* 재귀에 대한 이해 + 연산한 적이 있다면 바로 return한다. (다이나믹 아이디어)

**4) 프로그래머스 입국 심사 문제**

배운점)

* 역으로 나누기 연산 사용하기 -> 전체 시간에 배열의 요소들을 나누어 합하면 몇 명을 심사할 수 있는지가 나온다.
* Start, mid는 최소 최대가 있고 왔다 갔다 하는 흐름으로 결정할 수 있는 문제여야한다.
* 최소를 구하는 문제라면 (정확히 정해진 값이 아닌) mid = n 대신에 >=n 으로 로직을 짜면 최적의 해를 구할 수 있다.

**0822**

**1) 의상**

배운점)

* 딕셔너리 사용법 : 딕셔너리 value 안에 list를 넣을 수 있다. (append도 가능)

**2) 공유기 설치**

배운점)

* 아이디어는 맞았으나 구현 (어떤 인덱스를 return 할지 어려움.)이 부족

**3) 디스크 컨트롤러 (프로그래머스)**

잘한점)

* 생각 그대로 구현했더니 풀렸다! 큐에 넣고 빼고!

**4) 체육복 (프로그래머스)**

배운점)

* 쉬운 문제었는데 예외 케이스를 못찾음.
* 중복을 체크할 때는 set 연산을 사용해도 좋다!

**5) 타겟 넘버 (프로그래머스)**

* 그래프가 아니더라도 dfs를 적용할 수 있다! (모든 것을 다 계산해야 하는 경우라면)

0823

**1) 가장 먼 노드 (프로그래머스)**

배운점) graph와 visited의 분리, 배열 입력 방법

잘한점) bfs를 혼자서 구현했다! 양방향 그래프를 떠올렸다.

0824

2020 카카오 기출 문제 풀기

**1) 2020 카카오 신고 결과 받기 – sol (30분)**

배운점) 변수 이름 깔끔하게, 그리고 index 함수 유용하게 사용할 수 있음.

또한 꼭 dic을 사용해야할까?도 고민해보기.

잘한점) 28분안에 sol, 그리고 맞음. Set과 dic 사용하는 아이디어

**2) 2020 카카오 k진수에서 소수 개수 구하기- sol (30분)**

잘한점) 소수 구하는 로직, k진수 구하는 로직 잘 떠올림

**3) 2020 카카오 주차요금계산- sol (30분)**

배운점) 문제를 제대로 읽어야한다. 같은 차량이 왔다 갔다 할 수 있는 케이스를 생각하지 못했다. 적혀 있는 것은 반드시 무조건 파악해야한다.

**4) 2020 파괴되지 않은 건물 – sol**

**5) 2020 카카오 양궁 대회**

배운점) 사용해야할 배열의 크기가 작으면 겁먹지말고 완전 탐색도 고민하자.

0824

**1) 2020 카카오 양과 늑대**

배운점) 재귀에 대해서 고민. 문제 특성상 최적의 상태를 그리디적 접근으로 풀 수 없기 때문에 방문 표시 – dfs – 방문 표시를 지움으로 진행됨.

**2) 트리의 부모 찾기**

배운점) 그래프에서 부모란 한단계 먼저 방문한 노드라고 생각하면 된다.

잘한점) 현재 방문한 노드와, 이제 탐색할 노드를 부모-자식이라고 생각하여 맞는 로직을 짰다.

0827

**1) 택배배달과 수거하기**

배운점) 스택구조(배열 거꾸로) 하는 로직

잘한점) 그리디라고 판단해 먼 거리부터 탐색해야한다고 생각

**2) 이모티콘 할인행사**

배운점) 완전 탐색 아이디어 떠올리자!

잘한점) 중복조합을 생각한 아이디어, 3중으로 반복문이 복잡했는데 로직을 잘 생각했다.

**0828**

배운점) 문제를 똑바로 읽자. 대각선 케이스가 있는지 생각도 못했다.

잘한점) dfs 로직 빨리 생각하고 잘 구현했다.

**0831**

**1) 백준 신입사원**

배운점) 생각(글, 흐름)을 코드 로직으로 생각하는 방법

**2) 백준 오큰수 문제**

배운점) 우선 순위를 항상 쓸 필요는 없다. 스택으로 해결할 수 있었다.

잘한점) 로직을 잘 생각했다. (시간 문제와 상관 없이)

**3) 크게만들기**

배운점) 배열 문제에서는 항상 Index 잘 생각하기. 예외 생각 전에 케이스 분류하기.

**0901**

**1) 백준 쇠막대기**

잘한점) 로직 생각 잘했다 (스택 사용하기), 케이스 분류 잘했다 (잘랐을 때 수 + 원래 막대기 수)