코드암기용

def quick\_sort():  
  
 if len(arr) <= 1:  
 return arr  
 pivot = arr[len(arr) // 2]  
 lesser\_arr, equal\_arr, greater\_arr = [], [], []  
 for num in arr:  
 if num < pivot:  
 lesser\_arr.append(num)  
 elif num > pivot:  
 greater\_arr.append(num)  
 else:  
 equal\_arr.append(num)  
 return quick\_sort(lesser\_arr) + equal\_arr + quick\_sort(greater\_arr)

코테 풀면서 고려해야 하는 점. – 배열을 쓸 것인지, 정렬을 해야 하는 지, while문이 필요한지 ?

해쉬 – 중복을 비교할 때 사용됨.

key -> 정수화- 해시함수라고도 함 (배열의 index번호) -> 배열을 만듬 -> list

문제점

– 여러 key를 해시함수를 했는데 값이 같을 경우, 배열방이 같을 경우.

구현해야 하는 메소드

* Key의 문자열을 아스키코드화해서 다 더함 -> 이게 해시코드 함수
* Index는 key를 % size로 나눈 값을 찾아내는 함수
* 충돌여부 확인 메소드 – 충돌일 경우 링크드리스트로 저장.

코딩문제 리뷰

1. 소수찾기 문제
   1. 문제내용
      1. Numbers 배열이 주어짐. 길이는 1이상 7이하, 요소는 0~9까지의 숫자.
      2. 각 배열을 조합해서 소수를 만듬. 예) 1,7이 주어지면 17, 71, 1, 7을 조합할 수 있음. 이중 1은 소수취급을 안함으로 이 배열에서 만들 수 있는 소수는 총 3개
      3. 배열에서 만들 수 있는 소수를 리턴.
   2. 해결풀이
      1. 모든 조합을 구하는 재귀함수 작성
2. def purmutation(arr, prefix, \_list):  
    if len(arr) == 0:  
    if prefix != None:  
    \_list.append(prefix)  
    return \_list  
    else:  
    for i in range(0, len(arr)):  
    \_arr = arr[0:i] + arr[i + 1:]  
    \_prefix = prefix + arr[i]  
    if len(\_arr) >= 1:  
    purmutation(\_arr, '', \_list)  
    purmutation(\_arr, \_prefix, \_list)
   * 1. 셋을 통한 중복제거 후, 소수판별 시작.
     2. 소수판별 후에, 소수를 카운트 및 제출