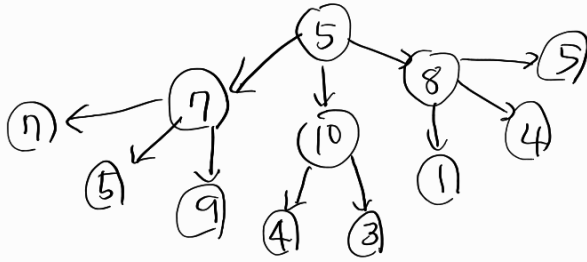


## 그리디 (탐욕적) 알고리즘

현재 상황에서 다음 단계 좋은 것만 고르는 방법으로 정답성 분석 (최적의 해를 구할 수 있는지)

### [문제상황]

주된 노드로부터 시작하여 거쳐갈 노드 값을 최대로 만들고 싶음.



<문제> 거스름돈 (종권), 최대한 적은 수의 지폐로 지불하는 최소 값

1000을 지불해야 할 때 500 → 100 → 50 → 10 순서로 지불이 좋다.

↳ 큰 단위가 항상 작은 단위의 배수이기 때문에

최적의 종류가 4라고 할 때,  $O(K)$  → 최적의 종류만큼 for문이 반복된다.

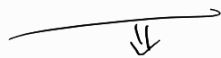
### <문제 27 1이 될 때까지>

- 어떤 수  $N$ 이 1이 될 때까지 다음의 두 과정 중 하나를 반복적으로 선택하여 수행,
- 단, 두 번째 연산은  $N$ 이 4로 나눌 때만 선택

1.  $N$ 에 1을 뺌
2.  $N$ 을 2로 나눴다.

Ex)  $N=17, k=4$ , 1번 과정 1번  $N=16$

2번 과정 2번  $N=1$



전체 실행 → 3번

$N$ 과  $K$ 가 주어질 때,  $N$ 이 1이 될 때까지

1번 혹은 2번의 과정을 수행해야 하는

최소 횟수를 구하시오.

<문제3> 곱하기 우선 처리하기.

[0]	[1]	[2]	[3]	[4]
0	2	9	8	4

그냥 곱셈 +, \* 비교해서 계산.

2번째 0이 안들어오면 곱셈 아닐때는 곱셈.

[0]	[1]	[2]	[3]	[4]
0	2	9	8	4.

1) 0을 앞단 넣고.

2) 0과 2중에서 0이나 1이 있으면 곱셈 아니면 곱셈

3) 2와 9중에서 //

4) (8과 8중에서 //

5) (18x8)과 4중에서 //

<12 314> 도둑가 길드

4명            2 3    1    2 2

↓ sort  
[0] [1] [2] [3] [4]  
3   2   2   2   1

1. 3이벤트가 3이벤트, 인원수 5이면 0~2까지  
1그룹

2. 2이벤트, 인원수 2이벤트 3~4까지 2그룹

3 <= 5 일때.

group=1

i=2

2 <= 5 일때..

구현: 머릿속에 있는 알고리즘을 소스코드로 바꾸는 과정.

problem  $\rightarrow$  thinking  $\rightarrow$  solution.

구현이 어려운 풀이를 떠올리게 하지만 소스코드로 옮기기 어려운 문제.

예시) ① 알고리즘은 간단해서, 코드가 지나칠만큼 길어지는 문제

② 실수연산을 다하고, 특정 소수점까지 출력해야 하는 문제

③ 문제별로 특정한 기호에 따라서 끝이 달라져야 하는 문제

④ 적절한 라이브러리를 찾아서 사용해야 하는 문제

(2차원 행렬 사용하기..)

열 (Column)

행 (row)

(0,0)	(0,1)	(0,2)	...
(1,0)			
(2,0)			
(3,0)			

(방향 벡터 사용하기..)

동 북 서 남

$$dx = [0, -1, 0, 1]$$

$$dy = [1, 0, -1, 0]$$

(221):  $\langle \text{행렬} \rangle$

C   P   U   D

$$dx = \begin{matrix} 0 & 0 & -1 & 1 \end{matrix}$$

$$dy = \begin{matrix} -1 & 1 & 0 & 0 \end{matrix}$$