

그리디(Greedy)

작성자 : 201907010 김민철

목 차

1. 그리디(탐욕적인) 알고리즘이란?

2. 그리디 알고리즘 사용 조건

3. 그리디 알고리즘 사용법

4. 실제 사용 예시

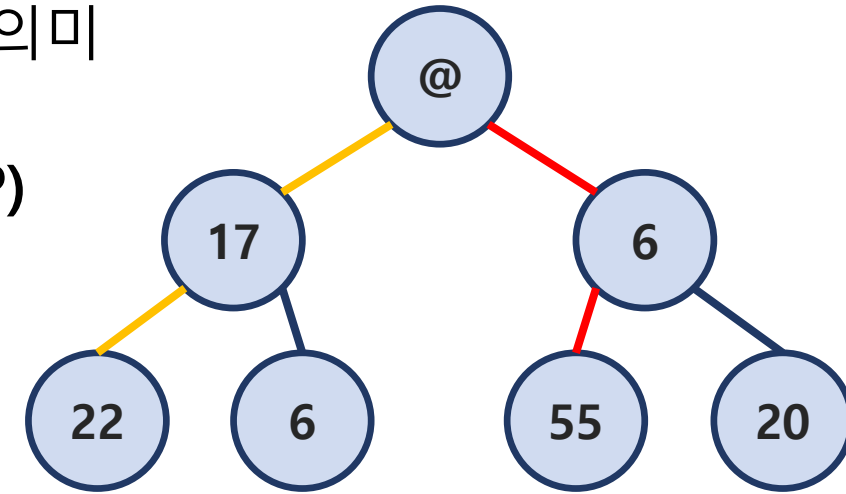


> 그리디(탐욕적인) 알고리즘이란?

#현재 상황에서 가능한 최선의 선택을 고르는 알고리즘을 의미

#부분 문제가 서로 독립되는 상황에서 **동적 프로그래밍(DP)**
보다 빠르게 문제를 해결하기 위해 사용

#만약 옆의 그림과 같은 상황에서 @지점을 시작으로
가장 큰 수를 찾아가야 하는 상황이 주어진다면 **노란
선과 같은 선택**을 하는 알고리즘이 그리디 알고리즘
(실제 답은 빨간 선과 같은 선택을 해야 함)



#그리디 알고리즘은 선택하는 시점에선 최선이지만 **최종적으로 최적해가 아닐 가능
성이 존재**하므로, 사용 가능한 **조건을 만족**하는지 먼저 확인해야 함

> 그리디 알고리즘 사용 조건

#탐욕스러운 선택 조건

→ 앞의 선택이 이후의 선택에 영향을 주어서 안됨

#최적 부분 구조 조건

→ 문제에 대한 최종적인 해결 방법이 곧 부분 문제의 해결 방법과 일치

#위의 조건을 만족하지 못하더라도 속도가 빠르다는 장점을 이용하여 **근사값을 찾기 위한 알고리즘**으로 사용은 가능

> 그리디 알고리즘 사용법

1. 선택 절차 : 현재 상태에서 최적해를 선택
2. 적절성 검사 : 선택된 해가 문제의 조건을 만족하는지 검사
3. 해답 검사 : 문제가 해결되었는지 검사, 만약 해결 되지 않았다면 선택 절차부터 다시 과정을 반복

> 그리디 알고리즘 사용법

#예를 들어 최대한 적은 개수의 동전을 거슬러 받기 위한 알고리즘을 만들 때 그리디 알고리즘을 적용한다면, 다음과 같은 과정으로 해결

1. 선택 절차 : 동전 개수를 줄이기 위해 가장 비싼 동전을 선택
2. 적절성 검사 : 1번 절차에서 고른 동전들을 모두 합쳤을 때 거스름 돈보다 비싼 지 검사 → 비싸다면 마지막 동전을 한단계 더 싼 동전으로 변경
3. 해답 검사 : 동전들의 합이 거스름 돈과 일치하는지 검사
→액수가 부족하다면 1번으로 돌아가서 반복

#이 문제에서 각 선택은 이전의 선택과는 관계없이 남은 거스름 돈 중에서 최대한 채울 수 있는 동전의 수를 선택하기 때문에 **탐욕스러운 선택 조건을 만족**하며, 큰 동전부터 선택하는 방법이 결국 가장 적은 수의 동전을 선택하는 방법이기 때문에 **최적 부분 구조 조건 또한 만족**

> 실제 사용 문제 예시

A → B 성공

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	512 MB	22783	9525	7627	40.451%

문제

정수 A를 B로 바꾸려고 한다. 가능한 연산은 다음과 같은 두 가지이다.

- 2를 곱한다.
- 1을 수의 가장 오른쪽에 추가한다.

A를 B로 바꾸는데 필요한 연산의 최솟값을 구해보자.

입력

첫째 줄에 A, B ($1 \leq A < B \leq 10^9$)가 주어진다.

출력

A를 B로 바꾸는데 필요한 연산의 최솟값에 1을 더한 값을 출력한다. 만들 수 없는 경우에는 -1을 출력한다.

예제 입력 1 [복사](#)

```
2 162
```

예제 출력 1 [복사](#)

```
5
```

> 실제 사용 예시

```
1  #include <iostream>
2
3  using namespace std;
4
5  long long int A, B, n;
6  long long int res = 1;
7
8  long long int comp(long long int a, long long int b)
9  {
10     while (a != b)
11     {
12         if (a > b)
13         {
14             return -1;
15         }
16         if (b % 10 == 1)
17         {
18             res++;
19             b /= 10;
20         }
21         else if (b % 2 == 0)
22         {
```

```
22         {
23             res++;
24             b /= 2;
25         }
26     }
27     else
28     {
29         return -1;
30     }
31     return res;
32 }
33
34 int main()
35 {
36     ios_base::sync_with_stdio(false);
37     cin.tie(nullptr);
38     cout.tie(nullptr);
39
40     cin >> A >> B;
41
42     cout << comp(A, B);
43
44 }
```




END