

Part 2. Algorithm

2017년 12월 26일 화요일 오후 8:14

코딩 테스트를 위한 TIP

1. 칠판에 글쓰기를 시작하십시오.
2. 그것을 통해 이야기하십시오.
 - 문제를 풀다가 막혔다면 말하는 것이 한 가지 방법이 될 수 있다.
 - 가끔은 면접관에게 진행 상황에 대해 명확하게 말하는 것이 지금 문제에서 무슨 일이 일어나고 있는지 이해할 수 있는 계기가 될 수 있다.
 - 당신의 면접관은 당신이 그 생각을 추구하도록 당신을 방해 할 수 있다.
3. 알고리즘을 생각하세요
 - 상향식 접근법: 때로는 문제의 세부 사항을 검토하고 해결책이 당신에게 나올 것을 기대하는 것이 유용하다.
 - 하향식 접근법: 다른 알고리즘에 대해서도 생각해 볼 수 있으며 각각의 알고리즘이 당신 앞의 문제에 적용되는지 질문 할 수 있다.
 - 위와 같이 참조 프레임을 변경함으로써 즉각적인 통찰력을 얻을 수 있다.
4. 데이터 구조를 생각하십시오.
 - 상위 10 개의 데이터 구조가 실제 세계에서 사용하는 모든 데이터 구조의 99%를 차지한다.
5. 이전에 보았던 관련 문제와 해결 방법에 대해 생각해 보십시오.
 - 제시한 문제는 이전에 보았던 문제이거나 적어도 조금은 유사하다.
 - 문제가 제기되는 형태로 넘어지지 말아야 한다.
 - 핵심 과제로 넘어가서 과거에 해결 한 것과 일치하는지 확인한다.
6. 문제를 작은 문제로 분해하여 수정하십시오
 - 문제를 큰 문제의 하위 집합으로 축소하면 작은 부분부터 시작하여 전체 범위까지 작업을 진행할 수 있다.
 - 작은 문제의 구성으로 문제를 보는 것도 도움이 될 수 있다.
7. 되돌아오는 것을 두려워하지 마십시오.
 - 특정 접근법의 효과적이지 않다고 느끼면 다른 접근 방식을 시도할 때가 있다.
 - 유망한 생각이 들지 않는 접근법에 몇 분을 소비했다면, 백업하고 다른 것을 시도한다.

문제 해결을 위한 전략적 접근

코딩 테스트의 목적

- 궁극적으로 문제 해결능력을 측정하기 위함이며, "어떻게 이 문제를 창의적으로 해결할 것인가"를 측정하기 위함이다.

- 문제 해결 여부
- 예외 상황과 경계값 처리
- 코드 가독성과 중복 제거 여부 등 코드 품질
- 언어 이해도
- 효율성

접근하기

- 해당 문제에 대해 완벽하게 이해하는 것이 우선이어야 한다.
 - 문제를 공격적으로 받아들이고 필요한 정보를 추가적으로 요구한다.
- 추상화
 - 해당 문제를 익숙한 용어로 재정의하거나 문제를 해결하기 위한 정보를 추출한다.
- 알고리즘과 자료구조를 고민한다.
 - 추상화된 데이터를 기반으로 이 문제를 어떻게 해결할 지 계획을 세운다.
- 계획에 대한 검증을 해본다.
- 계획으로 문제를 해결한다. 해결이 되지 않는 경우 앞의 과정을 되짚어본다.

해결 방법 분류

동적 계획법(Dynamic Programming, DP)

- 재귀적으로 호출했을 때 중복적인 계산이 발생하는 경우, 중복된 부분은 caching하는 방법을 Memoization이라 한다.
- 모든 선택지를 고려해보고 그중 가장 좋은 것을 찾는 방법

탐욕법(Greedy)

- 각 단계마다 가장 좋은 방법만을 선택하여 해결하는 방법
- 동적 계획법보다 수행 시간이 빠르다.