



실전 알고리즘 0x04강 스택 1(수식의 괄호 쌍)

BaaaaaaaaaaaaaaaaarkingDog

목차



0x00 수식의 괄호 쌍이란?

0x01 문제 해결을 위한 관찰

0x02 문제 해결 방법

0x03 응용 문제 소개

0x00 수식의 괄호 쌍이란?



- 수식의 괄호 쌍 문제는 주어진 수식의 괄호 쌍이 올바른지 판단하는 문제입니다.
- 편의상 수식은 제외하고 괄호 쌍만 생각해봅시다.
- $(()) \rightarrow O$
- $(()) \rightarrow X$
- $))(((\rightarrow X$
- $((())) \rightarrow O$

0x00 수식의 괄호 쌍이란?



- 직접 눈으로 관찰하면 그냥저냥 올바른지 아닌지 판단을 할 수 있긴 하겠는데 과연 이 문제를 어떻게 컴퓨터로 해결할 수 있을까요?
- 놀랍게도 이 문제는 스택으로 $O(N)$ 에 해결할 수 있습니다.
- 코딩테스트 단골 문제는 아니지만 잊을만하면 형태를 살짝 바꾸거나, 조금 응용해서 나오는 문제입니다.

0x01 문제 해결을 위한 관찰



- 사람이 머릿속에서 푸는 과정을 컴퓨터에게 시킬 수 있도록 명료하게 정리를 해야 합니다.
- 다음 슬라이드로 넘어가기 전에 `((()))()` 이 문자열이 올바른 문자열인지 아닌지 어떻게 판단할 것인지를 충분히 고민해보세요.

0x01 문제 해결을 위한 관찰



- 가장 쉽게 착안할 수 있는 방법은 (와) 의 갯수를 세는 것입니다. 일단 둘의 갯수가 다르면 올바르지 않은 문자열일 것입니다.
- 그러나 둘의 갯수가 같아도 올바르지 않은 문자열일 수 있습니다. 반례 :) (
- 관찰력이 뛰어나시다면 모든 순간에) 의 갯수가 (의 갯수를 넘지 않으면 된다는 사실을 알아차릴 수도 있습니다.
- 붙어있는 () 를 계속 지웠을 때 문자열을 모두 없앨 수 있는지를 확인하는 방법도 있습니다.
- ((()) ()) → (() ()) → (()) → () → 끝

0x01 문제 해결을 위한 관찰



- 괄호가 한 종류일 때는 갯수를 세는 방법으로 해결할 수 있지만 여러 종류일 때는 그렇지 않습니다. 예시: $(\{\}) / (\{\})$
- 붙어있는 $()$ 혹은 $\{\}$ 를 지우는 방법은 여전히 잘 동작합니다.
- $(\{(\{\})\}\}) \rightarrow (\{\{\}\}) \rightarrow (\{\}) \rightarrow () \rightarrow \text{끝}$
- 이 방법을 배열로 구현할 경우 최대 $N/2$ 번 중간에 있는 원소의 삭제가 발생하기 때문에 $O(N^2)$ 이고 연결 리스트로 구현할 경우 $O(N)$ 입니다.
- 연결 리스트로 구현을 하는게 엄청 어려운건 아니지만 스택을 이용한 조금 더 간단한 방법이 있습니다. 이 방법을 고안하기 위해서는 하나의 관찰이 추가로 더 필요합니다.

0x01 문제 해결을 위한 관찰



- 추가적인 관찰 : “문자열을 앞에서부터 읽어나갈 때, 닫는 괄호는 가장 최근에 들어온 여는 괄호를 없애버리는 명령이라고 생각해도 된다.”
- 예시를 보면서 이해해봅시다.

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

{

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

{ (

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

{ ((

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

{ (((

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

{ ((

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

{ (({

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

{ ((

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

{ (

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

{ ((

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

{ (

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

{

0x01 문제 해결을 위한 관찰



문자열 = { ((() { }) ()) }

0x02 문제 해결 방법



- 앞에서 처리하는 과정을 살펴볼 때 떠오른 자료구조가 없었나요? 가장 최근의 것이 가장 먼저 나오는 자료구조, 즉 **스택**입니다.
- 여는 괄호가 나오면 스택에 넣습니다. 닫는 괄호가 나오면 우선 스택이 비어있는지를 확인합니다. 스택이 비어있으면 잘못된 괄호 문자열입니다. 스택이 비어있지 않은 경우 스택의 꼭대기를 살펴봅니다. 스택의 꼭대기가 짝이 맞는 괄호인 경우 pop을 하고, 짝이 맞지 않을 경우에는 잘못된 괄호 문자열입니다.
- 모든 과정을 끝낸 뒤 스택에 문자열이 남아있으면 잘못된 문자열입니다.
- BOJ 9012번 : 괄호 문제의 정답코드 : <http://boj.kr/c54d954d852c4b4fa9ea237056489555>
- 9012번 문제는 원래 ()만 있을 때에 대한 문제이나 정답코드는 () { } 가 있을 때에도 처리할 수 있도록 짜두었으니 참고하세요.

0x03 응용 문제 소개



- 괄호쌍 문제와 관련해서 몇 가지의 응용 문제에 대해 간단한 접근법을 알려드리려고 합니다. 문제는 늘 그렇듯 그룹 내의 문제집에 있으니 혼자 힘으로 먼저 도전해보고 싶으신 분들은 뒤의 내용을 더 이상 읽지 마세요.

0x03 응용 문제 소개



- BOJ 10799번 : 쇠막대기
- 괄호쌍을 볼 때 지금 보고있는 이 닫힌 괄호가 레이저를 의미하는지 쇠막대기를 의미하는지는 그 앞의 괄호가 열린 괄호인지 아닌지를 보면 알 수 있습니다.
- 레이저를 쏘 때 몇 개의 막대기가 잘려나가는지를 어떻게 알 수 있을까요?
- 쏘는 그 순간에 스택의 길이를 생각해 보세요.

0x03 응용 문제 소개



- BOJ 2504번 : 괄호의 값
- 일단 사람 손으로 푸는 과정을 생각해봅시다.
- 이제 그 과정을 스택 상에서 어떻게 옮겨낼 수 있을지 고민해봅시다.
- 스택에서 단순히 문자만 가지고 있는게 아니라, 값도 가지고 있어야 한다는 사실을 눈치채셨나요?
- 닫힌 괄호를 만나 스택에서 문자와 값을 빼낼 때, 무언가 계산을 한 뒤 그 결과를 어디에 넣어줘야 할까요?

0x03 응용 문제 소개



- BOJ 2493번 : 탑
- 괄호쌍과는 관련이 없는 스택 응용 문제입니다.
- 각 탑에 대해 나보다 높은 탑 중에 왼쪽으로 가장 가까이 있는 탑을 찾아야 합니다.
- 2 3 4 6 5 1 이 있을 때, 2 3 4는 6이 나온 이후로 절대 레이저 신호를 수신할 일이 없습니다.
- 6 또한 만약 그 뒤에 6보다 더 큰 탑이 등장한다면 그 이후로는 더 이상 신경쓸 필요가 없습니다.
- 스택에 어떻게 내림차순으로 값을 잘 두고, 나보다 크게 나올 때 까지 pop을 계속 한다면..?

강의 정리



- 스택의 괄호 쌍에 대해 공부했습니다.
- 스택을 응용한 문제들에 대해 살펴보았습니다.
- 접근법을 보고 풀었든, 보지 않고 풀었든 제시된 응용 문제 3개를 모두 풀어낼 수 있다면 스택과 관련한 코딩테스트 문제에서 어려움을 겪을 일은 없을 것입니다. 이것보다 더 어려운 수준으로 나오면 어차피 정말 극소수만 풀 수 있어요.
- 응용 문제의 풀이가 잘 이해가 가지 않는다고 해서 너무 스트레스 받지 마세요. 어려운게 당연한 겁니다. 다른 것들을 먼저 공부하고 나서 다시 살펴보시면 한결 나을거예요.