

字符串 Parse

请设计一个字符串 parse 函数，可以将输入的字符串分解为对应的树状结构，比如：

例子1：

```
let input = 'int';
let result = parse(input);
// result 结果为:
// {
//   type: 'int'
// };
```

例子2：

```
let input = 'Array<bool>';
let result = parse(input);
// {
//   type: 'Array',
//   typeArgs: [{
//     type: 'bool'
//   }]
// };
```

例子3：

```
let input = 'Array<Array<string>>';
let result = parse(input);
// {
//   type: 'Array',
//   typeArgs: [{
//     type: 'Array',
//     typeArgs: [{
//       type: 'string'
//     }]
//   }]
// };
```

例子4:

```
let input = 'Map<string, Array<bool>>';
let result = parse(input);
// {
//   type: 'Map',
//   typeArgs: [{
//     type: 'string'
//   }, {
//     type: 'Array',
//     typeArgs: [{
//       type: 'bool'
//     }]
//   }]
// };
```

同理，该 parse 函数可以分解如下任意情况，比如：

```
"int"
"string"
"bool"
"Array<int>"
"Array<Array<int>>"
"Array<Array<Array<int>>>"
"Map<string, Map<string, bool>>"
"Map<Map<string, bool>, bool>"
```