

# CST2013 2-3 Behtree

## 描述

行为树是一种控制AI运行的树，一般使用XML格式存储。我们将这棵树的XML格式简化成如下的上下文无关文法的BNF：

```
BehTree ::=
"<Behavior>"
RootNode
"</Behavior>"
RootNode ::= TreeNode
TreeNode ::= LeafNode | InnerNode
LeafNode ::= "<Node Name=" NameStr ">"
InnerNode ::=
"<Node Name=" NameStr ">"
    SonList
"</Node>"
SonList ::=
TreeNode
{TreeNode}
```

其中::=表示被定义为，其左边是定义的非终结符，右边为其表示。双引号内的是终结字符，引号外的是语法部分，这里出现的语法有A|B表示A或B，{A}表示A重复0到多次。特殊的是NameStr表示的一个带双引号的只包含大小写英文字母或0-9数字的字符串。

JSON格式是一种更简洁的序列化表示方法。以上定义的XML格式的树可以转化为对应的JSON格式：

```
BehTree ::= "{" "Behavior" ":" RootNode "}"
RootNode ::= TreeNode
TreeNode ::= LeafNode | InnerNode
LeafNode ::= "{" "Name" ":" NameStr "}"
InnerNode ::= "{" "Name" ":" NameStr ", " "SonList" ":" SonList "}"
SonList ::= "[" TreeNode {"," TreeNode} "]"
```

现在你作为一个程序员，需要编写一个程序，将输入的行为树的XML格式，转化为JSON格式输出。

## 输入

若干行，树的XML表示，输入的XML保证合法，其中包含一些多余的空格、tab和换行

## 输出

仅一行，树的JSON表示，不可包含多余空格和tab，孩子顺序与XML格式中一致。

## 输入样例

```
<Behavior>
  <Node Name="Root">
    <Node Name="Google">
      <Node Name="Search">
        <Node Name="Keyword"/>
      </Node>
    <Node Name="Map">
      <Node Name="Reduce"/>
    </Node>
    <Node Name="GFS"/>
  </Node>
</Node>
</Behavior>
```

## 输出样例

```
{ "Behavior": { "Name": "Root", "SonList": [ { "Name": "Google", "SonList": [ { "Name": "Search", "SonList": [ { "Name": "Keyword" } ] }, { "Name": "Map", "SonList": [ { "Name": "Reduce" } ] }, { "Name": "GFS" } ] } ] } }
```

## 限制

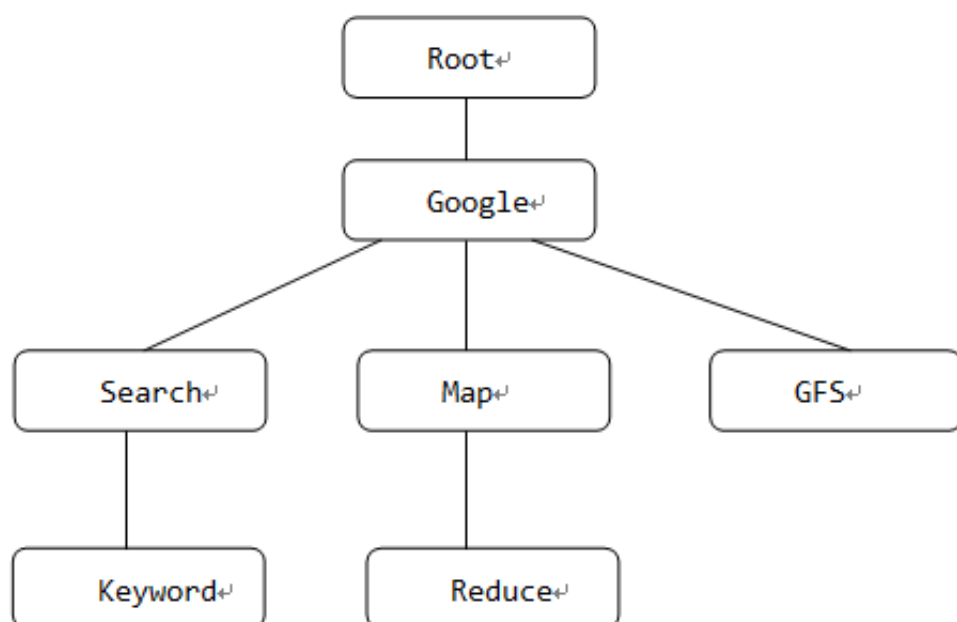
输入数据不超过1MB

时间限制：1秒

内存限制：128M

## 提示

样例的树形似：



---

UI powered by Twitter Bootstrap (<http://getbootstrap.com/>).

Tsinghua Online Judge is designed and coded by Li Ruizhe.

For all suggestions and bug reports, contact [oj\[at\]liruizhe\[dot\]org](mailto:oj[at]liruizhe[dot]org).