

面向对象第一单元第三次作业指导书

摘要

本次作业，需要完成的任务为包含简单幂函数和简单正余弦函数的导函数的求解。

问题

设定

首先是一些基本概念的声明：

- **带符号整数** 支持前导 0 的带符号整数，符号可忽略，如： $+02$ 、 -16 、 19260817 等。
- **因子**
 - **变量因子**
 - **幂函数**
 - **一般形式** 由自变量 x 和指数组成，指数为一个带符号整数，如： x^{+2} 。且，指数绝对值一律不得超过 10^4 。
 - **省略形式** 当指数为1的时候，可以采用省略形式，如： x 。
 - **三角函数** $\sin(x)$ ， $\cos(x)$ ，另外，本指导书范围内所有的词语“三角函数”，除非特别说明，否则一律包含且仅包含上述两个函数)
 - **一般形式** 类似于幂函数，由 $\sin(x)$ ， $\cos(x)$ 和指数组成，指数为一个带符号整数，如： $\sin(x)^{+2}$ 。同样的，指数绝对值一律不得超过 10^4 。
 - **省略形式** 当指数为1的时候，可以采用省略形式，省略指数部分，如： $\sin(x)$ 。
 - **常数因子** 包含一个带符号整数，如： 233 。
 - **表达式因子** 将在表达式的相关设定中进行详细介绍。不过，表达式因子不支持幂运算。
 - **嵌套因子** 本次作业将支持因子嵌套在三角函数因子内部，即一个因子作为另一个三角函数因子的自变量，例如 $\sin(x^2)$ ， $\cos(\sin(x))$ 以及 $\sin(\sin(\cos(\cos(x^2))))^2$ 等。但是不允许出现指数为变量的情况，指数依然只能是带符号整数，例如 $\sin(x)^{\sin(x)}$ 是不合法的，因为指数不是自变量。也不允许幂函数的自变量为除了 x 之外的因子，例如 1926^{0817} 是不合法的，因为幂函数的自变量只能为 x 。
- **项**
 - **一般形式** 由乘法运算符连接若干任意因子组成，如： $x * \cos(x) * x$ ， $\sin(x^2) * \cos(\sin(x))^{2 * x^{-2}}$ 等。
 - **项内因子** 不仅仅是同类因子
 - **特殊形式**
 - 第一个因子为常数因子 1 且其后跟着乘号的时候，可以省略该常数因子或表示为正号开头

的形式，如： $x^2 * x^{-1}$ 、 $+x^2$ 、 $+\cos(x) * \cos(x)$ 、 $\sin(x) * \cos(x)$ 。

- 第一个因子为常数因子 -1 且其后跟着乘号的时候，可以表示为负号开头的形式，如： $-x^2$ 、 $-\cos(x) * \sin(x)$ 。
- **表达式** 由加法和减法运算符等若干项组成，如： $(-1 + x^{233}) * \sin(x^2) \wedge 06 - \cos(\sin(x)) * 3 * \sin((x))$ 。此外，在**第一项之前**，可以带一个正号或者负号，如： $- -1 + x^{233}$ 、 $+ -2 + x^{1926}$ 。此处有几点注意：
 - **表达式因子**，表达式可以作为因子，其定义为被一对小括号包裹起来的表达式，即我们允许诸如 $(x * \cos(x))$ 这种式子的出现
 - **空串不属于合法的表达式**
- **空白字符** 在本次作业中，空白字符包含且仅包含 `<space>` 和 `\t`。其他的除了上述会用到的字符之外，均属于非法字符。

此外，值得注意的几点是：

- 带符号整数内不允许包含空白字符。
- 三角函数的保留字内不允许包含空白字符，即 `sin`，`cos` 关键字内不可以含有空白字符。
- 因子、项、表达式，在不与上两条矛盾的前提下，可以在任意位置包含任意数量的空白字符。
- 表达式因子可以递归嵌套，例如 $\sin((x^2 + \sin((1 + 3*x))))$
- 如果某表达式存在不同的解释方式，则只要有任意一条解释中是合法的，该表达式即为合法。
- 为了方便评测机评测，**我们限制的指数的绝对值不超过 10^4** ，超过此范围需要输出 **WRONG FORMAT!**
- **(重大更新)** 对于类似 $\sin((x-x))^{-1}$ 的输入，会存在除0的情况，对于评测是非常难处理的，故对指导书进行补充。为了保证向下兼容，我们限制**所有的指数必须严格大于0**，这一条作为基本限制，任何测试数据不得违背该限制，不需要对此输出 **WRONG FORMAT!**。此外，我们将 0^0 一概定义为 1（即c标准库和python中的定义）

描述

求导是数学计算中的一个计算方法，它的定义就是，当自变量的增量趋于零时，因变量的增量与自变量的增量之商的极限。

在本次作业中，我们要对输入的多项式进行求导计算，并输出它的导函数。

本次作业可能用到的求导公式有：

I. 当 $f(x) = a$ (a 为常数) 时， $f'(x) = 0$

II. 当 $f(x) = ax^b$ ($b \neq 0$) 时， $f'(x) = abx^{b-1}$

III. 当 $f(x) = \sin(x)$ 时， $f'(x) = \cos(x)$

IV. 当 $f(x) = \cos(x)$ 时， $f'(x) = -\sin(x)$

V. 当 $f(x) = \arctan(x)$ 时， $f'(x) = 1/(1+x^2)$ (曾经有，本次作业取消)

$$\text{VI. 链式法则: } [f(g(x))]' = f'(g(x))g'(x)$$

$$\text{VII. 乘法: 当 } f(x) = g(x)h(x) \text{ 时, } f'(x) = g'(x)h(x) + g(x)h'(x)$$

判定

输入格式

输入中，包含且仅包含一行，表示一个表达式

输出格式

关于输出，首先程序需要对输入数据的合法性进行判定

- 如果是一组合法的输入数据（即符合上述的表达式基本规则），则应当输出一行，表示求出的导函数。格式同样需要满足上述的表达式基本格式规则。
- 如果是一组不合法的输入数据，则应当输出一行 `WRONG FORMAT!`

判定模式

正确性判定

对于这次作业结果正确性的判定，在输出符合格式要求的前提下，我们采用和上一次完全一样的方式，可以直接跳至下一段落。具体如下：

- 在区间 $[-10, 10]$ 上，线性随机选取1000个数，设为 $\{x_i\} (1 \leq i \leq 1000)$
- 设输入多项式为 $f(x)$ ，其导函数为 $f'(x)$ （即正确答案，由sympy进行符号运算），将 $\{x_i\}$ 依次代入 $f'(x)$ ，得到结果 $\{a_i\}$
- 设待测输出的多项式为 $g'(x)$ ，将 $\{x_i\}$ 依次代入 $g'(x)$ 进行符号运算，得到结果 $\{b_i\}$
- 将数列 $\{a_i\}$ 和数列 $\{b_i\}$ 依次进行比较，判定每个数是否依次相等
- 如果全部相等，则认为该组输出正确，否则认为错误

综上，简而言之，你可以理解为：**只要是和标准结果等价的表达式（允许定义域上的点差异），都会被认定为正确答案。**

性能分判定

在本次作业中，性能分的唯一评判依据，是输出结果的有效长度。

有效长度定义为，输出结果去除所有的空白字符（`<space>`、`\t`）后的长度，设为 L 。

设某同学给出的正确答案的有效长度为 L_p ，所有人目前给出的正确答案的有效长度的最小值为 L_{min} 。

设 $x = \frac{L_p}{L_{min}}$ ，则该同学性能分百分比为：

$$r(x) = 100\% \cdot \begin{cases} 1 & x = 1 \\ -0.2984x^4 + 2.3111x^3 - 6.1699x^2 + 6.073x - 0.9158 & 1 < x \leq 3 \\ 0 & x > 3 \end{cases}$$

简单来说，就是**和第三次作业完全不一样**这样：

x	$r(x)$
1.0	100.0%
1.3	77.7%
1.5	60.0%
1.7	44.0%
2.0	26.5%
2.5	16.0%
3.0	0.0%

以及，由于格式错误的情况下，输出是固定的，所以实际上对于格式错误的数据点，只要被判定为正确即可获得100%的性能分。

值得注意的是，**获得性能分的前提是，在正确性判定环节被判定为正确**。如果被判定为错误，则性能分部分为0分。

互测相关

在互测环节

- 数据的最大长度为**60**。（请注意，这里不是有效长度，是去除右侧换行符后的总长度）。
- 对于符合表达式格式的输入数据，其正确导函数在 $[-10, 10]$ 的范围内不存在超过 10^{20} 的点。
- 对于所有的指数，必须是严格大于0的。

上述限制被定义为**数据基本限制**。在此范围限制内，不作其他任何限制。简而言之

- 如果是格式合法的数据，则被测程序应当给出正确的答案。
- 如果是格式不合法，但是满足上述数据基本限制的话，被测程序应当输出格式不合法情况下的结果。（即输出 WRONG FORMAT!）
- 如果不满足上述数据基本限制的话，则该数据将被系统忽略，不会对被测程序进行测试。
- 在公测中，也不会存在不满足数据基本限值的数据点。

对于基本限值的第二条，判定方式如下：

- 在区间 $[-10, 10]$ 上，线性随机选取5000个数，设为 $\{x_i\}$ ($1 \leq i \leq 5000$)
- 设输入多项式为 $f(x)$ ，其导函数为 $f'(x)$ （即正确答案，由sympy进行符号计算），将 $\{x_i\}$ 依次代入 $f'(x)$ ，得到结果 $\{a_i\}$ 。

- 如果满足 $\forall 1 \leq i \leq 5000, |a_i| \leq 10^{20}$ ，则输入数据判定为满足这一要求，否则判定为不满足。

样例

#	输入	输出	解释
1	1	0	显然。
2	$4*x+x^2+x+1$	$2*x+5$	幂函数与常量因子为同类因子
3	$4*x+x^2+x$	$4+2*x+1$	未合并同类项，但表达式依然等价。
4	$4x+x^2+x$	WRONG FORMAT!	$4x$ 不是合法项，应该写作 $4*x$
5	$- -4*x + x^2 + x$	$2*x+5$	$-4x$ 为合法项，且表达式第一项前也可以包含正负号。
6	$+4*x - -x^2 + x$	$2*x+5$	$-x^2$ 为合法项。
7	$+19260817*x$	19260817	显然。
8	$+ 19260817*x$	19260817	多项式第一项前可以带有正负号。
9	$+++19260817*x$	WRONG FORMAT!	项内有符号整数不可以包含空白字符。
10	$+++1$	0	合法，实际上三个加号分别对应的是表达式、项、带符号整数。
11	$+++x$	WRONG FORMAT!	不合法，用现有文法无法正确解释。
12	$++++1$	WRONG FORMAT!	因子里头只能省略第一个，且必须在包含多个因子时才可以省略。
13	$+++++++$	WRONG FORMAT!	显然啊，不要想着全部省略这种奇怪的东西啦。
14	$1926\ 0817\ * \ x$	WRONG FORMAT!	同上。
15	(空)	WRONG FORMAT!	空串不属于合法表达式。
16	$2*\sin(x)$	$2*\cos(x)$	常数因子与三角函数相乘
18	$-1*\cos(x)$	$\sin(x)$	同上
18	$23+\sin(x)*3$	$3*\cos(x)$	同上
19	$\cos(x)*\sin(x)^5+4*x^3$	$-5*\sin(x)^2+5*\cos(x)*\cos(x)+12*x^2$	显然
20	$43+4*x^3$	$12*x^2$	仅包括常数因子项加包括幂函数的同类因子项
21	$6*\sin(x)$	WRONG FORMAT!	'sin'间不能有空白字符

22	6*cos(x)	WRONG FORMAT!	'cos'间不能有空白字符
23	6*arctan(x)	WRONG FORMAT!	无法识别的函数名(arctan已经取消了)
24	1*x^2	2*x	显然
25	sin(cos(x))	-cos(cos(x))*sin(x)	三角函数作为三角函数因子
26	sin((2 * x)) * (cos(x)+1)	2*(cos(x) + 1)*cos((2*x)) - sin(x)*sin((2*x))	显然
27	sin((2 * x)) * (cos(x)+1)^2	WRONG FORMAT!	表达式因子不得带有幂运算
28	sin (x)	cos(x)	显然
29	sin (x)	cos(x)	还是显然
30	sin(x)^10000	10000*cos(x)*sin(x)^9999	显然
31	sin(x)^10001	WRONG FORMAT!	指数绝对值一概不得超过 10^4
32	sin(2*x)	WRONG FORMAT!	'2*x'不是因子
33	1926^0817	WRONG FORMAT!	幂函数的自变量只能为x

注意：由于本作业可被判定为正确的答案不唯一，所以本测试样例仅供参考，**仅保证正确性，不保证其为性能最优解。**

补充信息

关于评测

- 评测时，会自动忽略掉行末的空格以及文件末多余的回车。
- 对于输入，如果包含多行，则忽略第一行以后的内容即可。（由于不忽略第一行以后的内容导致的错误，一概后果自负）
- 类似的，对于输出结果，如果包含多行，则在评判的时候将忽略第一行以后的内容。（也就是说，你们可以在正文之后附加一些其他的信息以改善自己调试的体验。以及，由于不在第一行输出正确答案造成的错误，也一概后果自负）

一点点的提示

- Java内的原生整数类型有 long 和 int ，长度分别为64位和32位。
- 如果觉得上述数据类型不够用的话，可以百度一下Java内可以怎样快速处理这个问题。
- 在Java内，不建议使用静态数组。推荐使用 ArrayList 、 HashMap 、 HashSet 一类的数据结构，快速管理和调配手中无序的数据。
- 关于输入字符串的处理，推荐使用**正则表达式**。
- 这次作业，看上去似乎很难，其实找对了方法后并不难。关键思想是，**化整为零**，可以这样考虑

- 对于每一种函数（常数、幂函数、三角函数），建立类
- 对于每一种函数组合规则（乘法、加减法、嵌套），建立类
- 对于上述的两种类，均实现一个求导接口
 - 其中，第一种类，做法显而易见
 - 其中，第二种类，做法一样显而易见
- 通过上述两种类及其求导接口，把整个表达式构建为树结构，也就是讨论区大佬们说的，链式求导
- 对于秒掉正确性部分后，想要最大限度优化性能的夫佬同学，一样可以将上述的化整为零思想作为可行思路之一，设计算法。
- 好了，提示到此为止，祝大家玩的愉快。

一点点想说的话

- **不要重复造轮子！不要重复造轮子！不要重复造轮子！重要的事情说三遍**
- 我们鼓励大家通过Baidu、Google、Stackoverflow等方式自行学习和解决问题。
- 如果还有更多的问题，请到讨论区提问。但是**请善用讨论区**，并在此之前认真阅读包括但不限于课程要求文档、指导书、搜索引擎结果等的内容。
- 如果想要深入了解Java的一些内置数据结构的特性和原理，推荐以下方法：
 - 查阅官方文档
 - 阅读该部分源代码（另外，在Idea内，Ctrl+左键点击方法名、变量名、类名、包名，有惊喜）
- 这次的性能分占比很低（不超过5%，且算分方法很宽松），请大家把重点放在架构的设计上，以及如何正确完成功能上。性能部分，学有余力再去仔细研究，做一些力所能及的优化。