

导言

这份文档主要用来存放一些实际工作中碰到的实用的代码片段,可能包含 MATLAB、Python、C 和一些 \LaTeX 的小知识。由于我是一个化学工程的学生,同时也初涉编程未深,计算机专业知识难免会出点错误,欢迎指正。

如果有人想编译这份手册或想学习一下实现,请务必读以下说明。

字体设置,为了避免侵权,尽可能使用开源字体^①。

- Source Han Sans: <https://github.com/adobe-fonts/source-han-sans/tree/release>
- Source Han Serif: <https://github.com/adobe-fonts/source-han-serif/tree/release>
- Source Code Pro: <https://github.com/adobe-fonts/source-code-pro>
- PT Sans Narrow: <https://fonts.google.com/specimen/PT+Sans+Narrow>
- TeX Gyre: 有问题前往<https://www.ctan.org>获取,一般来说 TeX 发行版自带
- 等宽字体: 大多数等宽字体都是程序员使用的,开源居多,颇易获取。我常用 DejaVu Sans Mono, Fira Code 和 Source Code Pro 三种。

```
%% 字体设置
\usepackage{fontspec}
\setmainfont{Adobe Garamond Pro} % TeX Gyre Pagella
\setsansfont{TeX Gyre Heros}
\setmonofont{Source Code Pro} % Consolas, DejaVu Sans Mono
\setCJKmainfont[BoldFont={Source Han Sans SC},
  ↳ ItalicFont={KaiTi}]{Source Han Serif SC}
\setCJKmonofont{FangSong}
\setCJKsansfont{Source Han Sans SC}

%% 数学字体
\usepackage{unicode-math}
\setmathfont[math-style = ISO, bold-style = ISO]{TeX Gyre
  ↳ Pagella Math}

%% url 样式
\newfontfamily\urlfont{PT Sans Narrow}
```

编译环境设置,代码高亮环境由 minted 宏包提供(需要 Python 环境)。

代码测试环境,各种代码的运行环境为 MATLAB 2017a、Anaconda 4.4.0 (Python 3.6x)、Visual Studio 2017 community、MiKTeX 2.9 (各宏包均为最新)。

^① 西文主字体 Adobe Garamond Pro、楷体、仿宋暂时没有找到理想的替代方案

第一章 MATLAB

问题 1 如何遍历当前文件夹及其子文件夹中的全部文件

假设现在我们有这样一个文件夹 A，它含有一些文件和子文件夹 B、C、D.....，这些子文件夹有包含若干层子文件夹。我们需要将这个父文件夹（A）及其子文件夹（B、C、D.....）和孙文件夹中的所有文件名和其路径取出来。

如果你用的是 MATLAB 2016b 及其后面的版本，那真的太棒了！`dir()`函数已经支持遍历搜索了。尝试敲入：

```
dir_data = dir('**/*');  
dir_data([dir_data.isdir]) = []; % 去除所有文件夹
```

这将会返回一个包含文件信息的 struct，现在你可以任意操作这些 struct 了，随意拼接路径。解放大脑，哦也！

方便归方便，但是，一来肯定有大多数人使用的是 MATLAB 2016b 之前的版本，二来，解放大脑意味着我失去了一次独立思考的机会。

思考

对于实现方法^①，多层次的遍历，我们第一时间想到的肯定是递归。然后就是数据的存储了，`dir()`函数返回的是一个 struct，这个数据结构储存有文件的 name 和 folder，我们要充分利用这个数据结构。所以现在思路是，写一个递归函数，这个函数返回包含所有文件的 struct。

这个函数应对先处理父文件夹，获取文件和子文件夹，在获取子文件夹的过程中，我们需要去除 '.' 和 '..' 这两个特殊的文件夹。我们对获取的子文件夹再次调用该函数。如此，利用递归获取子子孙孙无穷尽文件夹的信息^②，最后函数返回存储有所有文件信息的 struct。现在，

^① 思路来源：[How to get all files under a specific directory in MATLAB?](#)

^② 其实这并不可能，因为递归是有栈高度限制的，调用函数压入栈，返回函数弹出栈，如果文件夹层次太深，一直压栈就会到达栈溢出警告的极限，例如 Python 的栈往往是 100 层，我想 MATLAB 的栈也大致如此，不会太高

你可以对这个结构体做你想做的事情。思路如算法1.1所示。

Algorithm 1.1: 遍历获取当前文件夹及其所有子文件夹中的文件名

Input: path
Output: struct of file information
Function *get_all_file_name(path):*
 get file and sub_dir information of current dir;
 storing file information;
 remove specific folder;
 for *first sub_dir to last sub_dir* **do**
 get next sub_dir;
 recursion \Rightarrow *get_all_file_name(path);*
 end
end

解

MATLAB 2016b 以上的版本我们可以用函数返回 struct, 这个数据结构包含 [folder, name, date, bytes, isdir, datenum] 六个字段的信息, 我们可以按自己意愿使用 folder 和 name 拼接出文件的完整路径。

```
% get all file name in current dir and sub dir, Compatible with
↳ MATLAB R2016b and newer
function file_list = get_all_file_name_R2016b_newer(path)

dir_data = dir(path);
file_list = dir_data(~[dir_data.isdir]); % file name of current
↳ dir

% get sub dir information
sub_dir = dir_data([dir_data.isdir]); % struct
dot_dir = ismember({sub_dir.name}, {'.', '..'}); % logical
sub_dir = sub_dir(~dot_dir); % struct, remove specific folder

% recursion
for i = 1:length(sub_dir)
    next_dir = fullfile(sub_dir(i).folder, sub_dir(i).name); %
    ↳ str
    file_list = [file_list;
    ↳ get_all_file_name_R2016b_newer(next_dir)]; % struct
end

end
```

2016a 及之前的版本 dir struct 信息并不包含 folder, 如果返回 struct, 将只有文件的 [name,

date, bytes, isdir, datenum] 五个字段的信息，所以我们并不能根据函数返回的 struct 拼接出文件完整路径，我们需要自己将路径拼接成一个 cell，然后使用函数返回 cell。

```
% get all file name in current dir and sub dir, Compatible with
↳ MATLAB R2016a and older
function file_list = get_all_file_name_R2016a_older(path)

% file name of current dir
dir_data = dir(path);
file_list_struct = dir_data(~[dir_data.isdir]);
file_list = fullfile(path, {file_list_struct.name})';

% get sub dir information
sub_dir = dir_data([dir_data.isdir]); % struct
dot_dir = ismember({sub_dir.name}, {'.', '..'}); % logical
sub_dir = sub_dir(~dot_dir); % struct, remove specific folder

% recursion
for i = 1:length(sub_dir)
    next_dir = fullfile(path, sub_dir(i).name); % str
    file_list = [file_list;
        ↳ get_all_file_name_R2016a_older(next_dir)]; % struct
end

end
```

总结

dir()函数遍历整个 F 盘共 2 万余文件文件大约需要 1.555823s。我们实现的递归函数遍历 F 盘文件大约需要 3.703009s。慢是慢了点，但我们成功运用了递归解决问题，不是吗？

问题 2 title 中英文标题

第二章 Python

第三章 算法

问题 1 简单算法