
Algorithm 1: Simulation-optimization heuristic

Data: current period t , initial inventory I_{t-1} , initial capital B_{t-1} , demand samples

Result: Optimal order quantity Q_t^*

```
1  $r \leftarrow t$ ;  
2  $\Delta B^* \leftarrow -\infty$ ;  
3 while  $\Delta B \leq \Delta B^*$  and  $r \leq T$  do  
4    $Q \leftarrow \arg \max_{Q \geq 0} \Delta B_{t,r}^Q(I_{t-1}, B_{t-1})$ ;  
5    $\Delta B \leftarrow \Delta B_{t,r}^Q(I_{t-1}, B_{t-1}) / (r - t + 1)$ ;  
6   if  $\Delta B \geq \Delta B^*$  then  
7      $Q^* \leftarrow Q$ ;  
8      $\Delta B^* \leftarrow \Delta B$ ;  
9   end  
10   $r \leftarrow r + 1$ ;  
11 end
```

用LaTeX优雅地书写伪代码——Algorithm2e简明指南



薰风初入弦

上海交通大学 计算机科学与技术博士在读

关注他

465 人赞同了该文章

更新于2020/8/10: 感谢 @纯粹 大佬的补充, Algorithm2e本身不支持Do-While结构(支持的是While-Do), 需要自行定义。具体的定义方法请见最后一个例子

1. 准备

该工具包的使用手册下载地址: mlg.ulb.ac.be/files/algorithm2e.pdf

```
\usepackage[ruled,linesnumbered]{algorithm2e}
```

使用宏包时中括号的参数含义:

- `ruled` 是让标题显示在上面, 否则算法的标题则在下面。
- `linesnumbered` 让算法中显示行号。
- 还可以添加 `boxed`, 让算法排版时插入在一个盒子里。

2. 基本语法

书写代码时也有一些专门的命令:



代码	含义
<code>\</code>	在行末添加分号，并自动换行
<code>\caption{}</code>	插入标题
<code>\KwData {输入信息}</code>	效果为：“Data:输入信息”
<code>\KwIn {输入信息}</code>	效果为：“In: 输入信息”
<code>\KwOut {输出信息}</code>	效果为：“Out: 输出信息”
<code>\KwResult {输出信息}</code>	效果为：“Result: 输出信息”
<code>\For {条件}{循环语句}</code>	for 条件 do 循环语句 endo
<code>\If {条件}{肯定语句}</code>	if 条件 then 肯定语句 end
<code>\While {条件}{循环语句}</code>	while 条件 then 循环语句 end
<code>\tcc{注释}</code>	<code>/* 注释 */</code>
<code>\tcp{注释}</code>	<code>//注释</code>
<code>\elf {条件}{肯定语句}{否定语句}</code>	if 条件 then 肯定语句 else 否定语句 end

知乎 @薰风初入弦

强烈建议知乎支持Markdown表格

EX:

- 如果你不想让你的伪代码叫做 'Algorithm 编号'，可以使用 `\renewcommand{\algorithmcfname}{算法名}` 命令来修改。
- 除了`\If`、`\Else`、`\Elseif`之外，还有`\uIf`、`\uIf`、`\uElse`、`\Else`、`\uElseif`、`\Elseif`等命令，他们的区别在于
 - `\If`、`\Else`、`\Elseif`都是会以`end`结尾
 - `\uIf`、`\uElse`、`\uElseif`，是不以`end`结尾的块级元素
 - `\If`、`\Else`、`\Elseif` 是不以`end`为结尾的行内元素
 - 在`If-else`结构中，`\elf` 自带`else`（即 `if` 和 `else` 共用一个 `end`），而只是用 `\If` 和 `\Else` 的话则会多出一个`end`给`Else`。

3. 例子

这些例子可以直接放在一个空的`.tex`文件中进行编译。

例子1

来自官方文档的，什么都没有的原始例子





```

\def\SetClass{article}
\documentclass{\SetClass}
\usepackage[lined,boxed,commentsnumbered]{algorithm2e}
\begin{document}
\begin{algorithm}[H]
  \SetAlgoLined
  \KwData{this text}
  \KwResult{how to write algorithm with \LaTeX2e }

  initialization\;
  \While{not at end of this document}{
    read current\;
    \eIf{understand}{
      go to next section\;
      current section becomes this one\;
    }{
      go back to the beginning of current section\;
    }
  }
  \caption{How to write algorithms}
\end{algorithm}
\end{document}

```

Data: this text
Result: how to write algorithm with $\text{\LaTeX}2\text{e}$
 initialization;
while *not at end of this document* **do**
 read current;
 if *understand* **then**
 go to next section;
 current section becomes this one;
 else
 go back to the beginning of current section;
 end
end

Algorithm 1: How to write algorithms 知乎 @薰风初入弦

例子2

现在，我们在输入输出中间加一点点间隔，然后给算法的某些行进行强调，再给if条件加上注释呢

```

\def\SetClass{article}
\documentclass{\SetClass}
\usepackage[linesnumbered,lined,boxed,commentsnumbered]{algorithm2e}
\begin{document}
\IncMargin{1em}
\begin{algorithm}
  \SetKwData{Left}{left}\SetKwData{This}{this}\SetKwData{Up}{up}
  \SetKwFunction{Union}{Union}\SetKwFunction{FindCompress}{FindCompress}
  \SetKwInOut{Input}{input}\SetKwInOut{Output}{output}

  \Input{A bitmap  $\text{\Im}$  of size  $\text{\W}\times\text{\L}$ }
  \Output{A partition of the bitmap}
  \BlankLine
  \emph{special treatment of the first line}\;
  \For{\mathfrak{i}\leftarrow 2 \ \KwTo \ \mathfrak{L}}{
    \emph{special treatment of the first element of line  $\text{\mathfrak{i}}$ }\;
    \For{\mathfrak{j}\leftarrow 2 \ \KwTo \ \mathfrak{W}}{\label{forins}
      \Leftarrow \FindCompress{\mathfrak{Im}[\mathfrak{i},\mathfrak{j}-1]}\;
      \Up \leftarrow \FindCompress{\mathfrak{Im}[\mathfrak{i}-1,\mathfrak{j}]}\;
      \This \leftarrow \FindCompress{\mathfrak{Im}[\mathfrak{i},\mathfrak{j}]}\;
      \If{\tcp*[h]{0(\Left,\This)==1}}{\Left compatible with \This}\label{lt}
      \lIf{\Left < \This}{\Union{\Left,\This}}
      \lElse{\Union{\This,\Left}}
    }
    \If{\tcp*[f]{0(\Up,\This)==1}}{\Up compatible with \This}\label{ut}
    \lIf{\Up < \This}{\Union{\Up,\This}}
  }

```

```

\tcp{\This is put under \Up to keep tree as flat as possible}\label{cmt}
\lElse{\Union{\This,\Up}}\tcp*[h]{\This linked to \Up}\label{lElse}
}
}
\lForEach{element  $e$  of the line  $i$ }{\FindCompress{p}}
}
\caption{disjoint decomposition}\label{algo_disjdec}
\end{algorithm}\DecMargin{1em}
\end{document}

```

```

input : A bitmap  $Im$  of size  $w \times l$ 
output: A partition of the bitmap
1 special treatment of the first line;
2 for  $i \leftarrow 2$  to  $l$  do
3   special treatment of the first element of line  $i$ ;
4   for  $j \leftarrow 2$  to  $w$  do
5      $left \leftarrow \text{FindCompress}(Im[i, j-1])$ ;
6      $up \leftarrow \text{FindCompress}(Im[i-1, j])$ ;
7      $this \leftarrow \text{FindCompress}(Im[i, j])$ ;
8     if  $left$  compatible with  $this$  then //  $O(left, this) == 1$ 
9       if  $left < this$  then  $\text{Union}(left, this)$  else  $\text{Union}(this, left)$ 
10    end
11    if  $up$  compatible with  $this$  then //  $O(up, this) == 1$ 
12      if  $up < this$  then  $\text{Union}(up, this)$  // this is put under
        up to keep tree as flat as possible
13      else  $\text{Union}(this, up)$  // this linked to up
14    end
15  end
16  foreach element  $e$  of the line  $i$  do  $\text{FindCompress}(p)$ 
17 end

```

Algorithm 1: disjoint decomposition

知乎 @薰风初△弦

例子3

实际上，咱们用的最多的格式（论文里常见）就还是这种：

```

\def\SetClass{article}
\documentclass{\SetClass}
\usepackage[ruled,linesnumbered]{algorithm2e}
\begin{document}
\begin{algorithm}
\caption{Simulation-optimization heuristic}\label{algorithm}
\KwData{current period  $t$ , initial inventory  $I_{t-1}$ , initial capital  $B_{t-1}$ , de
\KwResult{Optimal order quantity  $Q^{\ast}_t$ }
 $r \leftarrow t$ ;
 $\Delta B^{\ast} \leftarrow -\infty$ ;
\While{ $\Delta B \leq \Delta B^{\ast}$  and  $r \leq T$ }{ $Q \leftarrow \arg\max_{Q \geq 0} \text{De}$ 
 $\Delta B \leftarrow \Delta B^{\ast}_t(r)(I_{t-1}, B_{t-1})/(r-t+1)$ ;
\If{ $\Delta B \geq \Delta B^{\ast}$ }{ $Q^{\ast} \leftarrow Q$ ;
 $\Delta B^{\ast} \leftarrow \Delta B$ ;}
 $r \leftarrow r+1$ ;}
\end{algorithm}
\end{document}

```

Algorithm 1: Simulation-optimization heuristic

Data: current period t , initial inventory I_{t-1} , initial capital B_{t-1} , demand samples

Result: Optimal order quantity Q_t^*

```

1  $r \leftarrow t$ ;
2  $\Delta B^* \leftarrow -\infty$ ;
3 while  $\Delta B \leq \Delta B^*$  and  $r \leq T$  do
4    $Q \leftarrow \arg \max_{Q \geq 0} \Delta B_{t,r}^Q(I_{t-1}, B_{t-1})$ ;
5    $\Delta B \leftarrow \Delta B_{t,r}^Q(I_{t-1}, B_{t-1}) / (r - t + 1)$ ;
6   if  $\Delta B \geq \Delta B^*$  then
7      $Q^* \leftarrow Q$ ;
8      $\Delta B^* \leftarrow \Delta B$ ;
9   end
10   $r \leftarrow r + 1$ ;
11 end
```

知乎 @薰风初入弦

例子4

Algorithm2e本身不支持Do-While结构（支持的是While-Do），需要自行定义。不过自行定义并不难，因为宏包中内置了Repeat-Until结构，在Algorithm2e中是“宏指令（Repeat macros）”的一种^[1]

自定义宏指令

```
\SetKwRepeat{Do}{do}{while}
```

定义完之后，就可以在伪代码块中使用如下命令调用

```
\Do{<结束条件>}{<执行命令>}
```

完整例程：

```

\documentclass{article}
\usepackage[linesnumbered, ruled]{algorithm2e}
\SetKwRepeat{Do}{do}{while}%
\begin{document}

\begin{algorithm}[H]
  \KwData{this text}
  \KwResult{how to write algorithm with \LaTeX2e }
  initialization\;
  \While{not at end of this document}{
    read current\;
    \Repeat{this end condition}{
      do these things\;
    }
    \eIf{understand}{
      go to next section\;
      current section becomes this one\;
    }{
      go back to the beginning of current section\;
    }
    \Do{this end condition}{
      do these things\;
    }
  }
  \caption{How to write algorithms}
\end{algorithm}

\end{document}
```

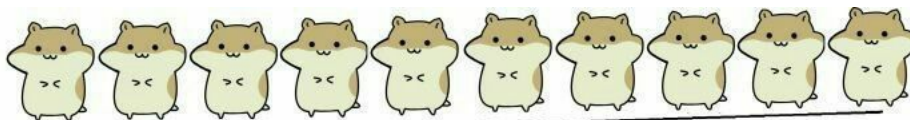
Algorithm 1: How to write algorithms

Data: this text

Result: how to write algorithm with L^AT_EX2_ε

```
1 initialization;
2 while not at end of this document do
3   read current;
4   repeat
5     do these things;
6   until this end condition;
7   if understand then
8     go to next section;
9     current section becomes this one;
10  else
11    go back to the beginning of current section;
12  end
13  do
14    do these things;
15  while this end condition;
16 end
```

知乎 @薰风初入弦



我们是寻找瓜子的仓鼠分割线

写在最后

LaTeX，说难也难，说不难也不难。

说它难，是因为其本身相当于一种较为独立的编程语言。熟练的掌握它，需要将以往我们在Word中“所见即所得”的写作思维彻底转变为“所想即所得”的思维。

但是，它也很简单。实际上，我从下载第一个CT_EX包，到本科第一篇论文写出来，只花了三天。LaTeX虽然涉及很多复杂与繁琐的设定，但是如果只是想粗略地入个门，只需要一个手把手带你配环境的小白向教程以及一个可以作为工具书查阅的工具书即可。（当然，系统学习LaTeX将会让你之后写论文的速度和版面美观度突飞猛进）

针对那些TeX小白用户，或者不太熟练的用户，我首先推荐我同门师兄小鱼学长写的教程~

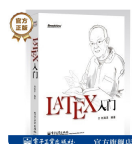
白小鱼：【小白向】LaTeX 中文入门

zhuatlan.zhihu.com



最后，LaTeX作为理工科科研人员的必备工具，熟练掌握可以极大地提高论文写作的效率。所以不论你是为了给日后宝贵的实验节省时间，还是为了更加优美地排版出想要的书籍，**拥有一本详尽且好懂的LaTeX教程，总比遇到问题了和无头苍蝇乱撞一样翻知乎和博客来得高效。**

因此，我在这里推荐我当时入门LaTeX使用的教材，可以在学有余力的时候系统性学习也可以在遇到问题时作为高效查阅的工具书~现在购买还有半价优惠哦！



全网低价的LaTeX教程大全~

京东

¥ 119.00

去购买 >

参考文献



[zhuanlan.zhihu.com/p/10...](#)
[blog.csdn.net/robert_ch...](#)
[blog.csdn.net/yq_foreve...](#)

参考

1. [^] <https://tex.stackexchange.com/questions/212301/do-while-loop-in-algorithm2e>

编辑于 08-10

[科研](#) [论文写作](#) [LaTeX](#)

文章被以下专栏收录



薰风的计算机科学家之路
深度学习，神经网络压缩，计算机视觉有关的随笔

关注专栏

推荐阅读



LaTeX引擎、格式、宏包、版大梳理
Rabby... 发表于Te

17 条评论

⇌ 切换为时间排序

写下你的评论...

😊

白小鱼

08-03

噢噢，学会了，正需要这个

👍 1

薰风初入弦 (作者) 回复 白小鱼

08-03

👍 1

阳光四格

08-03

[赞]

👍 2

▲ 赞



YuanJun

08-04

畅少牛逼 (破音)

👍 1



LdBeth

08-05

没有 goto, 差评(

👍 赞



薰风初入弦 (作者) 回复 LdBeth

08-05

讲真, 会有人在伪代码里用goto的吗。。

👍 赞



LdBeth 回复 薰风初入弦 (作者)

08-05

比如 Donald E. Knuth 就认为合理使用 goto 能让代码更清晰

👍 赞

展开其他 1 条回复



Pluveto

08-06

这.....这不是 vb 吗 (

👍 赞



薰风初入弦 (作者) 回复 Pluveto

08-07

没...没写过vb

👍 赞



纯粹

08-07

可以在导言区自定义一些新的语法, 比如do...while...结构

<https://tex.stackexchange.com/q/212301>

👍 赞



薰风初入弦 (作者) 回复 纯粹

08-08

哇, 我明天学习一下补充这个文章, 十分感谢你的补充

👍 赞



lomoyi

08-08

最近刚好.在看

👍 赞



suger的老爹

08-10

论文里用过。

👍 赞



武超

08-15

请教版主, 伪代码的符号说明有介绍的吗? 很多符号我都不知道什么意思。

👍 赞



学弱獠

08-18

nb

👍 赞



洗脚看电视

08-20

厉害

👍 赞