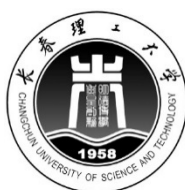


编号_____



长春理工大学
王大珩
Changchun University of Science and Technology

本科生毕业论文

毕业论文中文题目

The Subject of Graduation Project

学 生 姓 名	_____ 姓名 _____
专 业	_____ 专业 _____
学 号	_____ 23333 _____
指 导 教 师	_____ 指导老师 _____
学 院	_____ 学院 _____

二〇二一年 六月

毕业设计（论文）原创承诺书

1. 本人承诺：所呈交的毕业设计（论文）《毕业论文中文题目》，是认真学习理解学校的《长春理工大学本科毕业设计（论文）工作条例》后，在教师的指导下，保质保量独立地完成了任务书中规定的内容，不弄虚作假，不抄袭别人的工作内容。

2. 本人在毕业设计（论文）中引用他人的观点和研究成果，均在文中加以注释或以参考文献形式列出，对本文的研究工作做出重要贡献的个人和集体均已在文中注明。

3. 在毕业设计（论文）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

4. 本人完全了解学校关于保存、使用毕业设计（论文）的规定，即：按照学校要求提交论文和相关材料的印刷本和电子版本；同意学校保留毕业设计（论文）的复印件和电子版本，允许被查阅和借阅；学校可以采用影印、缩印或其他复制手段保存毕业设计（论文），可以公布其中的全部或部分内容。

以上承诺的法律结果将完全由本人承担!

作者签名： 年 月 日

摘要

这是一篇长春理工大学本科生毕业论文 L^AT_EX 模板?

这是一篇长春理工大学本科生毕业论文 L^AT_EX 模板.

这真的是一篇长春理工大学本科生毕业论文 L^AT_EX 模板?

这确实是一篇长春理工大学本科生毕业论文 L^AT_EX 模板.

关键词：关键词 1 关键词 2 关键词 3

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Key Words: Word1; Word2; Word3;

目 录

摘要	I
Abstract	II
第 1 章 第一章标题	1
1.1 第一小节	1
1.1.1 第一小小节	1
1.1.1.1 第一小小小节	1
1.1.2 第二小小节	1
1.1.2.1 第二小小小节	1
1.1.2.2 第四级标题	1
1.2 第二小节	1
1.3 第三小节	2
第 2 章 新的大节	3
第 3 章 参考文献和交叉引用	4
3.1 参考文献	4
3.1.1 交叉引用	4
3.2 公式这么用	4
第 4 章 用图和表的示例	6
第 5 章 代码块	7
5.1 代码块原样输出	7
5.2 图的使用	10
5.3 表的使用	12
5.4 列表的使用	12
致谢	13
参考文献	14
附录 1 第一个附录	15
1.1 附录可以有小节	15
1.1.1 小小节	15
1.1.1.1 附录中也可以有小小节	15

第 1 章 第一章标题

这是小四号的正文字体，段间距 1.5 倍

通过空一行实现段落换行，仅仅是回车并不会产生新的段落

也可以通过\par命令来新起一段

1.1 第一小节

1.1.1 第一小小节

1.1.1.1 第一小小小节

1.1.2 第二小小节

1.1.2.1 第二小小小节

1.1.2.2 第四级标题

段落 这是一个带有顶头标签的段落这是一个带有顶头标签的段落这是一个带有顶头标签的段落这是一个带有顶头标签的段落这是一个带有顶头标签的段落这是一个带有顶头标签的段落

小段落 只是一个带有缩进标签的段落只是一个带有缩进标签的段落只是一个带有缩进标签的段落只是一个带有缩进标签的段落只是一个带有缩进标签的段落只是一个带有缩进标签的段落

1.2 第二小节

本模板已经引入伪加粗和伪斜体，这样就不需要对应的粗体和斜体字体也能生成需要的效果，就像下面这样

宋体加粗

宋体斜体

宋体粗斜体

请注意，使用加粗和斜体时，请与字体名称一同使用，否则会自动将粗体匹配为黑体，斜体匹配为楷体，就像下面这样

正常显示宋体

加粗后变为黑体

斜体后变为楷体

第 2 章 新的大节

新的大节会自动出现在新的一页上

第 3 章 参考文献和交叉引用

3.1 参考文献

这是一个参考文献引用的范例 [1]，你可以随时使用两种不同的样式 [1] 或者^[1]

可以同时引用多个文献 [1-3]

这样可以添加一个不标注的参考文献引用

多引用另一情况 [1, 2, 4]

这样可以添加所有 bib 文件中的参考文献

参考文献使用了 thebibliography，没有用原模板的 bib 生成，这样做方便直接复制黏贴对应格式的引用。但是这样会导致 bibtex.exe 的一些错误或者警告提示，无视就好 (因为懒得改了，如果您有兴趣，可以更新扔 github 上)。

3.1.1 交叉引用

本模板已经重写了 hyperref 宏包的 \autoref 命令，方便引用章节、公式和图表。

比如说第 3 章、3.1 节和 3.1.1 节就引用了本章节，3.1.1.0 节和 3.1.1.0 节引用了之前的两个段落。显然段落因为没有序号，引用结果和上一节的需要相同，因此建议使用段落“段落”和段落“小段落”。

3.2 公式这么用

在文中引用公式可以这么写： $a^2 + b^2 = c^2$ 这是勾股定理，他还可以表示为 $c = \sqrt{a^2 + b^2}$ ，还可以让公式单独一段并且加上编号

$$\sin^2 \theta + \cos^2 \theta = 1 \quad (3-1)$$

还可以通过添加标签在正文中引用公式，如式 (3-1) 或者式 (3-1)。我们还可以轻松打出一个矩阵

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 11 & 22 & 33 & 44 \end{bmatrix} \times \begin{bmatrix} 22 & 24 \\ 32 & 34 \\ 42 & 44 \\ 52 & 54 \end{bmatrix} \quad (3-2)$$

或者多个带编号的公式

$$f_1(x) = 12x^2 + 36x + \sin x \quad (3-3)$$

$$f_2(x) = \sqrt{3}x^3 + 3x \quad (3-4)$$

指定某个不带编号

$$\begin{aligned} f(x) &= 4x^2 + 5x^2 \\ &= 9x^2 \end{aligned} \quad (3-5)$$

对齐等号

$$f(x) = 4x^2 + 5x^2 \quad (3-6)$$

$$= 9x^2 \quad (3-7)$$

左边大括号

$$\begin{cases} f(x) = \frac{\log_2(3x^2)}{e^x} \\ g(x) = \sqrt{3^{2x}} \end{cases} \quad (3-8)$$

来个泰勒公式：

$$f(x) = \frac{f(x_0)}{0!} + \frac{f'(x_0)}{1!}(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n + R_n(x)$$

Cauchy 不等式：

$$\left(\int_E |f(x)g(x)| d\mu \right)^2 \leq \int_E |f(x)|^2 d\mu \int_E |g(x)|^2 d\mu$$

麦克斯韦方程组：

$$\oint_S \mathbf{D} \cdot d\mathbf{A} = Q_{f,S} \quad (3-9)$$

$$\oint_S \mathbf{B} \cdot d\mathbf{A} = 0 \quad (3-10)$$

$$\oint_{\partial S} \mathbf{E} \cdot d\mathbf{l} = -\frac{\partial \Phi_{B,S}}{\partial t} \quad (3-11)$$

$$\oint_{\partial S} \mathbf{H} \cdot d\mathbf{l} = I_{f,s} + \frac{\partial \Phi_{D,S}}{\partial t} \quad (3-12)$$

对附录中的公式进行引用式 (① -1)

以上

$$\frac{123}{2}$$

第 4 章 用图和表的示例

第 5 章 代码块

5.1 代码块原样输出

```

print('hello world')

1 // Keywords Search
2 // 网络流上流传最广的AC自动机模板题，问你目标串中出现了几个模式串
3 // 如果一个结点是单词末尾的话out标记为true,在search的时候对于每个结点都
   向fail指针找
4 // 找到out为true的就将其标记为false,且ans++
5 #include <cstdio>
6 #include <iostream>
7 #include <string.h>
8 using namespace std;
9 #define maxn 10000000
10 struct node
11 {
12     int cnt; //是否为单词最后一个节点
13     node *next[26],* fail; // Trie 每个节点的26个子节点and失败指针
14     node()
15     {
16         cnt=0;
17         fail=NULL;
18         for( int i=0; i<26; i++)
19             next[i]=0;
20     }
21 }*root;
22 node *q[maxn]; //构造失败指针的bfs队列
23 char keyword[55]; //输入单词,模式串
24 char str[1000010]; //需要查找的主串
25 int head, tail; //队列头尾指针
26 void insert (char *s,node *p) //插入字符串函数
27 {
28     int i=0,k;
29     while(s[i])
30     {
31         k=s[i++]-'a';

```

```

32         if(p->next[k]==NULL)
33             p->next[k]=new node();
34             p=p->next[k]; // 指针移动至下一层
35     }
36     p->cnt++; // 单词结尾节点count + 1做标记
37 }
38 void build_AC(node *root) // 构造失败指针
39 {
40     head=0;
41     tail =1;
42     q[head]=root;
43     node *temp,*p;
44     while(head<tail) // bfs 构造Trie树的失败指针
45     {
46         temp=q[head++];
47         for( int i=0; i<26; i++)
48         {
49             if(temp->next[i]) // 判断实际存在的节点
50             {
51                 // root下的第一层节点的失败指针都指向root
52                 if(temp==root)
53                     temp->next[i]-> fail=root;
54                 else
55                 {
56                     // 依次回溯该节点的父节点的失败指针
57                     // 直到某节点的next[i]与该节点相同, 则
58                     // 把该节点的失败指针指向该next[i]节点
59                     // 若回溯到 root 都没有找到, 则该节点的失败指针指向 root
60                     p=temp->fail; // temp 为节点的父指针
61                     while(p)
62                     {
63                         if(p->next[i])
64                         {
65                             temp->next[i]-> fail =
66                                 p->next[i];

```

```

66                                     break;
67                                     }
68                                     p=p->fail;
69                                     }
70                                     if (!p)temp->next[i]-> fail=root;
71                                     }
72                                     //每处理一个点,就把它的所有儿子加入队列
73                                     //直到队列为空
74                                     q[ tail ++]=temp->next[i];
75                                     }
76                                     }
77                                     }
78 }
79 int query(char *str ,node *root)
80 {
81     int ans=0,k,len= strlen ( str );
82     node *p=root;
83     //i为主串指针, p为匹配串指针
84     for( int i=0; i<len; i++)
85     {
86         k=str [ i]- 'a';
87         //由失败指针回溯寻找,判断str[i]是否存在于Trie树中
88         while (! p->next[k]&&p!=root)
89             p=p->fail;
90         p=p->next[k]; //找到后p指向该节点
91         if (!p)p=root; //指针仍为空, 则没有找到与之匹配的字符
92         //指针重新回到根节点root,下次从root开始搜索Trie树
93         node *temp=p; //匹配该节点后,沿其失败指针回溯,判断其他节点
94                             是否匹配
95         while(temp!=root&&temp->cnt!=-1) //匹配结束控制and未被访问控制
96         {
97             ans+=temp->cnt;
98             temp->cnt=-1; //标记已访问
99             temp=temp->fail; //回溯失败指针继续寻找下一个满足
100                             条件的节点
101         }
102     }
103     return ans;

```

```

102 }
103 void del(node *root)
104 {
105     if (root==NULL)return ;
106     for ( int i=0;i<26;i++)del( root->next[i] );
107     delete (root);
108 }
109 int main()
110 {
111     int t,n;
112     scanf( "%d",&t);
113     while(t--)
114     {
115         root=new node();
116         scanf( "%d",&n);
117         for ( int i=0; i<n; i++)
118         {
119             scanf( " %s",keyword);
120             insert (keyword,root);
121         }
122         build_AC(root);
123         scanf( " %s",str);
124         printf ( "%d\n",query(str , root ));
125         del( root );
126     }
127 }

```

5.2 图的使用

其使用方法如下：

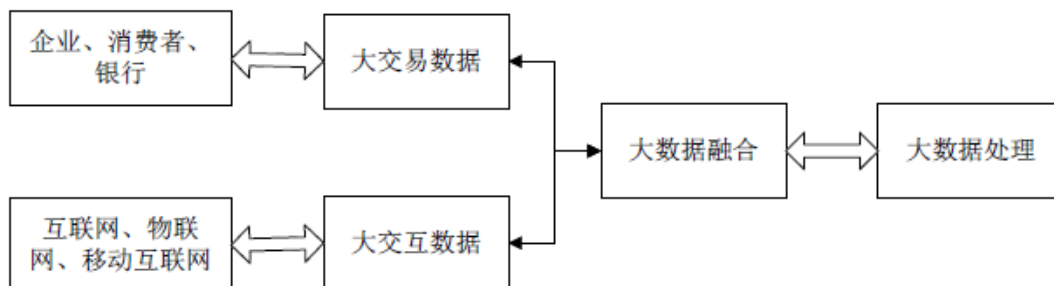


图 5-1 大数据信息处理框架

同时也可以引用该图片例如：图 5-1 。请注意 `generalfig` 第一个参数是标题，第二个参数是引用。

对附录上的图片引用图 ① -1

5.3 表的使用

作为论文，推荐使用三线表进行排版。所谓三线表，即在标题前有横线，标题后有横线，表格最后还有横线，其他地方无线。当然这不是死规定，也可以根据需要在合适的地方加线。

本文定义了新的可变长度左中右（LCR）格式，LCR 三个格式会根据表格宽度的设定自行控制宽度，且其宽度相等，方便设置和页面相同宽度的表格。但该功能需要使用 tabularx 做表。

表 5-1 某校学生升高体重样本

序号	年龄	身高	体重
1	14	156	42
2	16	158	45
3	14	162	48
4	15	163	50
平均	15	159.75	46.25

当然你也可以引用表格，就像这样：表 5-1 。引用附录表格表 ① -1

5.4 列表的使用

这是一个计数的列表

1. 第一项
 - (a) 第一项中的第一项
 - (b) 第一项中的第二项
2. 第二项
3. 第三项

这是一个不计数的列表

- 第一项
 - 第一项中的第一项
 - 第一项中的第二项
- 第二项
- 第三项

致谢

感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师
感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师
感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师
感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师
感谢老师感谢老师感谢老师

感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师
感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师
感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师感谢老师
感谢老师

参考文献

- [1] Arandjelović R, Zisserman A, Three things everyone should know to improve object retrieval, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012: 2911-2918.
- [2] Philbin J, Chum O, Isard M, et al. Lost in quantization: Improving particular object retrieval in large scale image databases, Computer Vision and Pattern Recognition, 2008. CVPR 2008, IEEE Conference on, IEEE, 2008: 1-8.
- [3] 王静康, 张凤宝, 夏淑倩, et al. 论化工本科专业国际认证与国内认证的“实质性” [J]. 高等工程教育研究. 2014, 5: 1-4
- [4] 陈剑. 上博简《民之父母》 “而得既塞於四海矣” 句解释 [EB/OL]. 简帛研究网站. <http://www.bamboosilk.org/Wssf/2003/chenjian03.htm>

附录 1 第一个附录

这里是附录环境，其中的 section、subsection、subsubsection 已经变为附录的样式，并且会以这种样式加入目录中

1.1 附录可以有小节

1.1.1 小小节

附录公式引用使用带圈数字

$$\nabla F = \left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z}\right)$$

(① -1)

1.1.1.1 附录中也可以有小小节

附录中的图片：

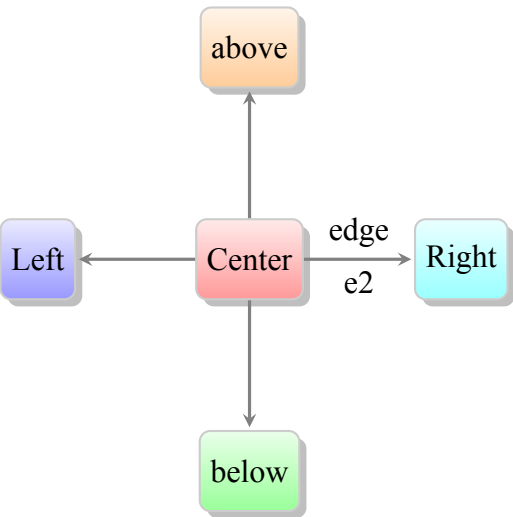


图 ① -1 Tikz 画流程图

附录中的表格：

表 ① -1 LaTeX Table Generator

a	s	d	f	g
3				
t				