

# 计算机学术论文写作经验总结

Summary of Paper Writing in Computer Science



2019 年 7 月 7 日

This page intentionally left blank.

# 目录

<b>1 前言(Preface)</b>	<b>7</b>
1.1 什么是学术论文?	7
1.2 本文的目的	8
<b>2 计算机论文结构(Structure of Research Papers in Computer Science)</b>	<b>9</b>
2.1 标题和作者列表 (Title and Author List)	9
2.2 摘要 (Abstract)	10
2.3 概述 (Introduction)	13
2.3.1 表述逻辑	13
2.3.2 四类重要的研究问题	13
2.4 相关研究 (Related Work)	15
2.5 基础知识 (Preliminaries)	17
2.6 新模型 (Model), 或者新概念 (Concept)	18
2.7 设计与实现 (Design and Implementation)	20
2.7.1 自顶而下(Top-Down)法	20
2.7.2 由浅入深法	21
2.7.3 抛砖引玉法	21
2.8 分析 (Analysis)	22
2.9 实验结果 (Experimental Validation)	23
2.9.1 实验环境搭建	23
2.9.2 实验内容设计	24
2.9.3 实验制图	24
2.10 总结及未来计划 (Conclusion and Future Work)	32
2.11 致谢 (Acknowledgement)	33
2.12 参考文献列表 (Bibliography)	33
<b>3 常见问题 (FAQ)</b>	<b>34</b>
3.1 如何寻找一个合适的研究问题或者研究方向?	34
3.2 如何搜索国际文献信息?	34
3.3 如何阅读一篇论文?	35
3.4 会议与期刊的区别	35
3.5 会议的CFP(Call For Paper)	37
3.6 计算机科研领域分类	37
3.7 什么是SCI/EI索引期刊	38
3.8 什么是影响因子(Impact Factor)	39
3.9 论文提交管理系统EasyChair、Microsoft CMT和EDAS	39
3.10 Microsoft Word与L <sup>A</sup> T <sub>E</sub> X之争	39
3.11 攻读硕士和博士期间应该学什么?	40
3.12 研究人员应具备那些计算机技能?	41
3.13 如何解决Font Not Embedded问题?	41



3.14	Regular Paper, Short Paper, Demo Paper 和Poster的区别	43
3.15	论文写作对英语写作能力要求有多高?	43
3.16	如何写一封完美的Cover Letter?	44
3.17	如果从其他领域“借鉴”一个方法, 相关研究(Related Work)章节应该如何组织呢?	45
3.18	如何在阅读论文的过程中发现新的研究问题?	45
3.19	如何应对会议/期刊论文被拒?	47
3.20	科研人员需要了解哪些会议/期刊信息?	48
<b>4</b>	<b>L<sup>A</sup>T<sub>E</sub>X常用的例子</b>	<b>49</b>
4.1	环境搭建	49
4.2	文章结构	49
4.3	文件结构	51
4.4	生成文章PDF版本	51
4.5	章节	52
4.6	图	53
4.6.1	单栏单图格式	54
4.6.2	跨栏单图格式	54
4.6.3	单栏双图并排格式	55
4.6.4	跨栏多图格式	57
4.7	表格	59
4.7.1	单栏横向表格	59
4.7.2	单栏纵向表格(一)	59
4.7.3	单栏纵向表格(二)	61
4.7.4	跨栏表格	63
4.7.5	合并单元格	64
4.8	数学模式与公式	64
4.8.1	数学模式	64
4.8.2	公式	65
4.9	算法	70
4.9.1	algorithm包	71
4.9.2	algorithm2e包	72
4.10	脚注	75
4.11	文献	76
4.12	符号表(Symbol Lists)	78
4.12.1	希腊字母表(Greek Alphabet)	78
4.12.2	函数名表(Function List)	79
4.12.3	数学符号表(Mathematical Symbols)	80
4.12.4	操作符表(Operator List)	81
4.12.5	箭头符号表(Arrow Symbols)	84
4.12.6	特殊字母表(Others)	85



<b>5 附录 (Appendix)</b>	<b>86</b>
5.1 会议与期刊排名 (Conference and Journal Rankings)	86
5.1.1 中国计算机学会(CCF)国际学术会议和期刊目录	86
5.1.2 中文核心期刊目录	86
5.1.3 Computing Research and Education Association of Australasia (CORE)	86
5.1.4 计算机会议和期刊排名（非官方）	86



This page intentionally left blank.

# 1 前言(Preface)

本文主要介绍计算机领域学术论文的相关知识，以及作者本人在研究工作中积累的一些撰写论文的经验。文章涵括了一系列的论文写作主题，陈述了论文写作的基本知识，并使用已发表文章为例，深入诠释了论文写作的技巧细节。本文中所有的内容和范例都是作者在历年研究过程中的结晶汇集。范例主要来自于作者已发表的文章，希望能够给初入研究的工作者一些帮助和启发。

本文着重介绍的是计算机学科领域的国际会议或者国际期刊论文。这些论文被称为 Research Paper 或者 Regular Paper；长度一般为 8-14 页，使用英文撰写，单栏或者双栏排版。综述性论文，毕业论文，短文，demo 论文不在本文讨论范围之内。

本文全部内容发布于<http://www.littlewaterdrop.com>网站上，供有心学者免费学习借鉴。网站还提供免费的pdf格式文件下载。笔者在写作过程中，兢兢业业，力求完整无瑕。然而由于笔者水平有限，文中难免出现疏忽、遗漏或者错误之处，尚期读者不吝指正。本文所有的变更和勘误会及时在该网站和pdf文件中实时更新。若对本文有任何的建议和意见，请与我们联系。邮件地址为：[support@littlewaterdrop.com](mailto:support@littlewaterdrop.com)。

本文倾注了作者大量的心血，并受版权保护，请勿转载。

## 1.1 什么是学术论文？

在开始介绍计算机学术论文之前，先了解一下论文存在的必要性和使用方法有助于理解论文结构。众所周知，做研究是一项困难的工作。一项成果的取得往往需要多名研究人员经长时间不间断的（广义上的）努力和合作。这些研究人员可能分布于世界各地，也可能出生于不同时代。那么，他们使用什么方法进行有效的沟通交流呢？

从科研工作的信息交流与传递方面来看，论文是他们有效交流的一种重要方式。一篇论文记录了研究人员在某一问题上获得的成果以及经验。这些成果和经验会被其他研究人员借鉴，完善，以及发展。这些成果和经验的借鉴者可能是刚刚走上科研岗位的初级研究人员，也可能是多年从事科研工作的专家。为了满足不同背景的研究人员的需求，一篇出色的论文需要将各方面的研究信息完整的诠释清楚。这些信息包括（但不限于）研究目的(Motivation)，背景调查(Survey)，研究方法(Method)，研究结果(Result)，以及未来研究计划(Future Plan)等。

另一方面，从论文的内容上看，一篇论文阐述了作者的一个观点或者假设 (Hypothesis)，以及使用一些初步数据或者实验结果来支撑这个观点和假设。发表一篇论文不仅仅是公开论文所阐述的研究成果，更是将所阐述的观点和假设的研究背景，理论依据和初步的实验结果公布于众，以期更多的科研人员投入这项研究，并进一步拓展这项研究。这里所说的拓展可包括：其他科研人员（1）进一步研究这个观点和假设；（2）尝试将该观点和假设应用和推广于其他应用场景；（3）进一步完善这个观点和假设，使之更加通用和有效。



基于对以上各个方面的考量，一篇论文质量的优劣往往由以下四个重要因素综合决定。

1. 重要性或者影响力（Impact），即该论文研究的问题是否非常重要，对相关的研究工作以及人们的生活是否有着重大的影响。
2. 新颖性或者独特性（Novelty）。在该论文中，是否提出了新的概念/系统/方法。这些新概念/系统/方法是未曾在其他文章中出现过的，是独一无二的。
3. 理论深度（Depth）。当确定了一个具体的研究问题后，接下来的一个问题是解决这个研究问题的难度。通常来讲，解决一个问题的难度往往取决于解决该问题使用的理论工具的难度。如果一个问题可以使用简单的方法解决，那么使用高难度的理论工具是完全没有必要的。但是，如果一个问题无法使用简单的方法解决，而一篇论文使用了较难（较复杂）的理论工具解决的话，那么，可以认为这篇论文具有较好的理论深度。
4. 提出新方法的有效性（Effectiveness）。这是对提出的新系统/方法的一种定量的验证。如果新系统/方法能够很好的改善问题现状的话，这个新系统/方法是非常值得推广的。（一些专注于理论研究的文章，也可通过定性的理论分析给出方法的有效性。）

## 1.2 本文的目的

计算机论文写作的方式和技巧多种多样，但是市面上或者网络上讲解这些经验和技巧的资料很少，而且很分散。这篇文章不仅仅是作为一名科研人员自身经验的总结，它还将散落在各处的资料汇聚在一起，希望能给其他科研人员以参考和借鉴。





## 2 计算机论文结构(Structure of Research Papers in Computer Science)

一篇计算机会议或期刊论文长度为8-14页不等（附录不算在内），约有8千-1万字。有些会议或期刊使用单栏排版，有些则使用双栏格式。一般会议论文有长度限制（8-12页），然而有些期刊有长度限制（12-14页），有些没有长度限制。

一篇优秀的论文不仅能清晰的表达作者的观点和方法，而且能帮助读者快速掌握文章的脉络以及重点。这是如何做到的呢？答案很简单，论文有其较为固定的框架，即较为固定的组成部分和逻辑顺序。

在大多数情况下，一篇论文由以下部分组成：标题(Title)，作者列表(Author List)，摘要(Abstract)，分类和关键字(Category and Keywords)，概述(Introduction)，相关研究(Related Work)，设计与实现(Design and Implementation)，实验(Experimental Validation)，总结(Conclusion)，致谢(Acknowledgement)，和参考文献(Bibliography)。有时论文还可能包含以下可选内容：背景知识(Preliminaries)，模型与新概念(Model and Concept)，分析(Analysis)，附录(Appendix)和作者简介(Biography)。以下逐一详细介绍每个部分。

### 2.1 标题和作者列表 (Title and Author List)

论文的标题需要能够准确并且充分地表达该论文所研究的问题以及核心思想。标题不宜过长，一般为一行或者两行。标题可以是一个句子或者一个短语，而且着重点应该是一个名词，即所研究的问题或者提出的新方法。例如，在如下的论文标题中明确地指出了该文章所研究的问题是“Skyline Query Processing”，而提出的新方法的核心思想是“MBR-Oriented”。

本例引用自 Ji Zhang, Wenlu Wang, Xunfei Jiang, Wei-Shinn Ku, and Hua Lu. An MBR-Oriented Approach for Efficient Skyline Query Processing. In Proceedings of the 35th IEEE International Conference on Data Engineering (ICDE), Macau SAR, China, 2019.

论文的作者列表常按照贡献程度依次列出作者的全名，所属单位和国家，以及电子邮件。常见的对论文的贡献包括（但不限于）：提出新的想法，设计与实现新的系统/方法/算法，参与实验实施，文档或论文编写，提供改进意见，以及资金资助等。

一篇期刊文章常常会在文章的作者中选择一位作为通讯作者(corresponding author)。有时，会议文章也会选择一位作者作为通讯作者。通讯作者常用于回答与该篇文章相关的问题。这些问题常常包括：解答读者对于文章的疑惑，更正文章的错误，说明各位作者对该篇文章的贡献等。



## 2.2 摘要 (Abstract)

摘要 是全论文的一个缩影，其重要性不言而喻。摘要应包含论文中各项重要的内容。例如，摘要应简明扼要的阐述研究问题的重要性，所提出新方法的新颖性，以及通过实验验证的新方法有效性的实验结果等。另外，正是因为摘要涵盖了论文各个重要的方面，许多读者或者审稿人会从摘要开始阅读，通过摘要内容来迅速地了解论文的内容，结构和逻辑，以初步评估论文贡献的大小。

以下两个例子中的摘要来自于作者2016年发表于IEEE TPDS的论文和2015年发表于ACM TOS的论文。这两篇论文的摘要逻辑清晰，表达层次分明，充分的展示了论文的主要贡献。由于版权原因，摘要的全文内容无法展示在例子中。文章的摘要可在例中提供的链接处获得。下面我们以这两篇摘要为例来逐句地分析摘要的写作方式，以及作者写作时的考量。例子中我们使用每句话开头的几个单词来标注句子序号，以便于读者查对。

例一引用自 Ji Zhang, Xunfei Jiang, Wei-Shinn Ku, and Xiao Qin. Efficient parallel skyline evaluation using MapReduce. IEEE Transactions on Parallel and Distributed Systems (TPDS), Volume 27, Issue 7, pp. 1996-2009, 2016. 例一的摘要原文可在网址 <https://ieeexplore.ieee.org/document/7219441> 处免费获取。

例二引用自 Ji Zhang, Xunfei Jiang, Xiao Qin, Wei-Shinn Ku, and Mohammed I. Alghamdi. Frog: A Framework for Context-Based File Systems. ACM Transactions on Storage (TOS), Volume 11, Issue 3, 2015. 例二的摘要原文可在网址 <https://dl.acm.org/citation.cfm?id=2720022> 处免费获取。

### 例一：

- ① This research develops an advanced ...
- ② Unlike existing parallel skyline approaches ...
- ③ In particular, we apply filtering techniques ...
- ④ In the second phase, local ...
- ⑤ To improve the parallel ...
- ⑥ The aggressive partition-aware ...
- ⑦ Recognizing the lack of studies ...
- ⑧ The presort process ...
- ⑨ Our experimental results show ...

注解：这篇是一个中等长度的摘要，共9句。①-②句表明该论文的目的和独特之处。③-⑧句进一步解释该论文的具体内容和特点，并主要围绕“为什么需要这个新系统”，“新系统设计了什么”，“新系统实现了什么”展开。⑨句简明扼要地阐述新方法的效果。其详细解释如下：

句①：开宗明义，明确本文是研究Skyline Query问题的一篇文章，采用的是基于MapReduce的框架。在这个框架下，本文设计了一个新的，基于两个阶段处理的并行方法。在这句话中，Skyline Query, MapReduce, Two-Phase这几个关键词很重要，也是全文的关键词。



句②：陈述该文章的新颖之处。Skyline Query的研究成果和基于MapReduce框架的研究成果非常多；但是，已发表的论文往往着重于改善和优化某一个并行Skyline Query处理的环节。此论文放眼于全局优化，将数据划分、过滤、和并行计算结合成为一个整体。所以，与该领域以往发表的论文相比较，最大的区别在于以往研究只做局部优化，而此文做出了全局优化。读完第二句，读者对该论文的轮廓及新方法应有一个大致的了解，并且应该能够抓住两个关键词：两个阶段，三个步骤。这三个步骤是数据划分，数据过滤，和并行计算。

句③-④：话锋一转，in particular, particularly或者specifically都用于表示后面将要讲述具体细节内容。在该摘要中，句③描述了第一阶段的实现细节，即新方法使用了过滤技术和以角度划分数据的方法。在完成数据划分的同时，新方法还去除了不合格的数据。句④阐述了第二阶段的工作步骤，即首先根据数据划分来并行处理各分区中的数据；然后，用一个全局的合并过程来计算并输出结果。这里我们可以看到，③-④句的内容实际上正对应着①-②句给出的关键信息（两个阶段和三个步骤）。读到这里，熟悉MapReduce工作原理的读者应对该新方法有了一个大致的认识，因为第一阶段和第二阶段都是一个典型的MapReduce计算的应用场景。

句⑤-⑥：在指出了较为详细的处理细节之后，⑤-⑥句的内容展示了此论文的独特之处。既然上述的两个场景都是MapReduce的典型场景，那么，这项研究的难点是什么呢？⑤-⑥句给出了第一个答案。由于第一阶段的数据划分直接影响着第二阶段的并行计算，因此，本文提出了两个partition-aware的过滤方法。这两个方法会尽量的将数据均匀的划分到各个分区。第一个方法的注意力放在过滤过程中最大的那个分区，并尽量使这个分区小一些。第二个方法则是按比例划分，即在过滤过程中，加大力度抑制较大分区数据量过快增长。在整个过滤过程中，分区的数据量是动态变化的。

句⑦-⑧：上述问题的第二个答案在⑦-⑧句中。⑦-⑧句解释了在第二阶段中的一个优化细节。第二阶段是一个典型的MapReduce应用场景。此文提出了一个partial-presort grid-based partition skyline算法，进一步提高第二阶段处理的性能。该方法利用了Mapper和Reducer之间固有的shuffling处理，以较小的代价将输入Reducer的数据做了一个预排序。这个预排序是一个偏序，而不是一个全序。

⑨句简单的总结了经实验验证的新方法的有效性。

读完这篇摘要后，读者应对整篇论文有了一个大致的了解。本篇论文在MapReduce框架下提出了解决Skyline Query的新方法。该方法包含两个处理阶段和三个处理步骤。为了进一步优化整个处理，新方法改善了第一阶段的数据划分，并利用shuffling提高了第二阶段的处理过程。带着这样的一个印象去读正文部分，读者应能更好的掌握整篇论文的要点以及逻辑脉络。反过来再看，本段对整篇文章印象的表述过程，也一一对应着摘要的表述顺序和层次结构。

## 例二：

① This article presents a framework ...

② Unlike existing informed-based ...

③ The framework can not only ...



- ④ Rather than propagating data ...
- ⑤ With Frog in place ...
- ⑥ Since the data consistency ...
- ⑦ We implement two prototypes ...
- ⑧ Inspired by an observation ...
- ⑨ To improve the performance ...
- ⑩ Our experimental results show ...

注解：例二也是一个中等长度的摘要，共10句。句①：在摘要开始处，表明该论文的研究目的和成果，即为了降低基于上下文文件系统或者应用程序的开发难度，本文开发了一个用于开发基于上下文文件系统（Context-Based File System）的框架。这里，上下文文件系统实际上只是一个概念，摘要后续提出来的两个实现的原型（BAVFS和BHVFS）是具体的基于上下文的文件系统。（基于上下文的系统一般工作于两个或者多个模式中。每个模式称为一个上下文。在某个具体的模式下，系统可以实现一些优化功能以提高系统整体的性能。）

句②-⑥：②-⑥在描述新方法特点的同时，介绍了概念上的实现细节。从概念上讲，基于上下文的文件系统并不是一个新的概念。然而，此文提出的文件系统却是全新的。它在文件系统中引入了视图（View）的概念；并使用一个视图来抽象出一个上下文环境。切换视图实际上就是切换上下文。因为新提出的系统是一个非常复杂的系统，它在很多方面与传统文件系统的工作方式不同，本篇摘要使用了五句话概述了新系统的内部（内部实现机制）和外部（对上层应用程序和开发人员）的优势，给读者描绘了一个大致的轮廓。

句⑦-⑨：在介绍完诸多优点之后，本文指出了两个典型的应用场景，并为其设计和实现了两个基于上下文的文件系统。这里作者是从对概念的描述转换到对实际应用的描述。在提出一个新概念或者系统之后，人们自然而然的会有两个疑问：这个新概念或者系统是否有着较广的应用面，以及这个新概念或系统应用的有效性如何。该文开发的两个基于上下文文件系统的原型就是为了回答这两个问题的。能开发两个原型，说明这个新概念或者系统有一定的应用面。而新系统的性能则由这两个系统原型的实验结果来表达展示。此处作者还进一步介绍了BAVFS可以将文件系统切分为随机读和顺序读两个场景，并使用Prefetching技术提升读的性能。BHVFS则将文件系统切分为读数据和写数据两个场景，并分别使用update-in-place和update-out-of-place 两种策略提升读写的性能。

句⑩：最后，作者指出，经实验验证，从新方法中获得的好处要远大于所付出的额外负担。

这篇摘要的写作具有一定难度。在论文的评审过程中，作者收到最多的评审意见均指出对于新系统的描述还不够全面。因为本论文提出来的的是一个全新的设计，它与传统的文件系统差别很大，所以，无论是新系统内部的工作原理，还是对外的接口与交互，都需要非常仔细的描述。在保持摘要简洁的原则上，需要完整清晰地表述研究目的，新系统工作原理，新系统的优势，新系统的应用场景以及实验结果显得较为困难。需要作者对每句话的表达和衔接反复推敲，琢磨。



## 2.3 概述 (Introduction)

### 2.3.1 表述逻辑

概述是论文正文开始的地方。在概述中，需要详细地说明论文的研究目的（或者动机）和研究成果。这两点缺一不可。合理的研究动机表明了这项研究的重要性，而研究成果展示了该研究工作非常成功，值得同行借鉴和推广。

那么，概述章节应该怎样组织，才能将研究目的和研究成效清晰且合理的表达出来呢？以下写作逻辑常常用于各种会议和期刊的论文中，读者不妨细细体会一下其中逻辑推导的过程。虽然，研究目的和研究成效在摘要中略有说明，然而，在概述章节中，这两点需要全面展开阐述，需要比摘要的描述更加细致，且更具有逻辑性。

概述章节可按照如下步骤展开叙述。

- 首先，强调目标问题的重要性。这里需要理解一个问题，即什么是重要的？通常来说，有以下四类问题是比较重要的。（1）具有影响力的问题；（2）能带来经济效益的研究问题；（3）社会热点问题；（4）能改善人们生活的问题。（还有更多重要的问题，这里并未一一列出。）在后续章节中会更详细的解释为什么这四类问题非常重要。
- 其次，简洁的分析一下该目标问题的现状。在提出一个研究问题之后，很自然的就会想到这个研究问题是否已被研究过，现在的研究水平如何。这一步其实是为了下一步做的铺垫，而且从逻辑上也是合理的。
- 然后，揭露当前研究方法的弊端，或者指出一些当前方法无法解决的问题。第三步是承接第二步的。在描述完最先进的方法之后，论文的逻辑转为批判最先进的方法，为论文提出新方法做准备。（试想，如果最先进的方法能够很完美的解决问题，那么，也就不需要该论文提出新方法了。）
- 承接上一步所暴露的目前研究所存在的问题，论文应详细论述新方法如何能够有效的改进或者解决这些问题。这里需要描述论文提出的新方法的主体思想。只不过在这里，论文需要将表达的重点放在如何解决暴露出来的问题，而不是简单的罗列新方法的内容。
- 最后，总结一下该研究的重要贡献。有时，虽然这些贡献会在上一步骤中详细解释。但是，这里需要再次总结一下这些贡献有助于读者或者审稿人快速定位描述论文贡献的段落位置。
- 结尾处，简单介绍一下后续章节的组织结构。这不仅是一个好的习惯，也可以帮助读者或者审稿人了解一下论文的框架结构。若论文篇幅较长，无法满足论文长度限制的情况下，这个步骤可以考虑省略。毕竟，这个步骤只是起到辅助的作用。删除后，并不太影响论文的质量。

### 2.3.2 四类重要的研究问题

研究问题是否重要是一个比较主观的问题，因人而异。然而，以下四类问题的重要性基本上





能够被绝大多数研究者接受。或者说，这些问题的重要性很难被反驳。这四类问题是

- 具有极大影响力的问题：影响力是个比较虚的概念，比较容易受到质疑。所以，如何将影响力以一种令人信服的方式表达出来至关重要。举例子是一种很常见并且很有效的表述方式。如果影响力体现在改变了人们的生活，那么，论文可以例举一些人们生活现状的例子，然后设想人们生活改变之后是什么样子，以此来做一个强对比就能较好的体现论文的影响力。如果影响力体现在攻克某项技术难题（尤其是基础研究）之后对其他研究的影响，那么，论文可以假想该技术难题已被攻克，并列出一一些受到影响的研究和方法。总之，论文需要将影响力具体化，实例化。一般来说，具有广泛影响力的论文较少见。绝大多数论文还是着眼于某一具体的研究问题，做出增量改进。
- 具有可在经济上取得较大回报的问题：经济上的改善是一个非常重要的驱动力。如果一项研究成果能够直接转化为经济成果，这样的研究成果是非常受研究人员和企业欢迎的。有很多方面的进步都可以直接或者间接的转化为经济回报。例如，（1）算法性能的提升。如果算法性能提升了，那么，意味着完成某项任务的时间缩短了，并且使用设备的开销也相应的降低了。（2）数据传输的有效性提高了。在网络资源非常稀缺的环境下（例如，卫星通讯），带宽是非常昂贵的。若能使用较少的带宽完成同样的任务，那么，实际上也是节约了带宽使用的成本。（3）节能。每一台运行的服务器都在消耗能量。如果能减少这些能量的消耗，那么，节省下来的能量开销也是一种经济回报。

这里给出一个与节能研究相关的例子。该例子选自一篇研究如何降低数据中心电力消耗的文章。据分析，美国数据中心的电耗占全美国总用电量的1.5%，全美国总电费为45亿美金。根据这组数字，我们可以得知，数据中心的用电量是非常大的（每年0.67亿美金）。如果某项技术能够减少数据中心的用电量（哪怕只是减少1%），那也是百万级美金的减少（每年）。所以，这项研究是非常有意义的。本例的具体内容可参考如下论文的概述部分。

本例引用自 Xunfei Jiang, Ji Zhang, Mohammed I. Alghamdi, Xiao Qin, Minghua Jiang, and Jifu Zhang. PEAM: Predictive Energy-Aware Management for Storage Systems. In Proceedings of the 8th IEEE International Conference on Networking, Architecture, and Storage (NAS 2013), Xi'an, China, 2013.

- 热点问题或者已被广泛开展研究的问题：这一点背后的逻辑是，如果已有很多研究者开展了某项研究工作，那么，这个研究一定是有意义的。当然，这里有个前提，即这些研究者在开展他们的研究工作之前已经做了充分的问题重要性的分析，并且在该问题上不断的有新研究者加入并提出改进意见。一般来说，出现在较好会议或者期刊上的论文都进行了充分的问题重要性分析，并且经过了同行的严格评审。

这里给出一个研究Skyline Query问题的例子。自从Skyline Query被作为一种操作（operator）加入数据库管理系统之后，先后有超过百位的研究人员投入到这项研究中。基于此问题发表的论文有几十篇之多。并且在此基础上，研究人员还探索了多种Skyline Query的变型问题。在下面的例子中，我们可以看到该论文并没有直接陈述问题的重要性，而是列出了已发表的相关成果。这不仅是一种问题重要性的表达，也是一种对于问题现状的描述。该文章作者相信读者或者审稿人不会质疑该问题的重要性。因为质疑本篇论文研究问题的重要性实际上就是质疑这几十篇已发表的论文所研究的问题的重要性，其结果不言而喻。

在这个例子中，已发表的方法被分为两类阐述：不使用索引（Index）的方法和使用索引的方法。因为已发表的方法数量实在太多，论文无法一一作出详细解释，于是只简单的列出其名



称。而这些方法在相关研究章节会做进一步的说明。本例的具体内容可参考如下论文的概述部分。

本例引用自 Ji Zhang, Xunfei Jiang, Wei-Shinn Ku, and Xiao Qin. Efficient parallel skyline evaluation using MapReduce. IEEE Transactions on Parallel and Distributed Systems (TPDS), Volume 27, Issue 7, pp. 1996-2009, 2016.

- 能改善人们生活的问题：这种问题也是研究的比较广、受到广泛关注的一类问题。这类问题离人们的实际生活较近，也比较容易展示其问题的重要性。在论述此类问题时，一个关键的诀窍就是例举一些人们日常生活中的例子，并且指出其中存在的问题。

下面的例子展示了如何来讨论一个日常生活的问题。此文研究的是在多个条件下如何正确做决策的问题，这个问题可以应用于城市规划领域。例如，这篇论文举了一个例子：如何寻找新小区的最佳位置的问题。假设，某开发商计划新建一个住宅区。为了吸引更多的买家，这个住宅区的选址需要考虑多个因素，比如，学区、到购物商场的距离、到公共交通的距离等因素。这是一个非常贴近人们生活的问题。如果这篇文章提出的方法能够较好的解决这个问题，那么，此文不仅能为开发商带来更多的经济收益，也能为住户提供更多的便利（因为这个小区的选址是最优的）。本例的具体内容可参考如下论文的概述部分。

本例引用自 Ji Zhang, Wei-Shinn Ku, Min-Te Sun, Xiao Qin, and Hua Lu. Multi-Criteria Optimal Location Query with Overlapping Voronoi Diagrams. In Proceedings of the 17th International Conference on Extending Database Technology (EDBT), Athens, Greece, 2014. 本例的全文可在如下网址处免费获取。

<https://openproceedings.org/2014/conf/edbt/ZhangKSQL14.pdf>

## 2.4 相关研究 (Related Work)

相关研究这一章节的重要性往往会被研究人员低估，因为本章节似乎只需要简单的罗列已有的相关工作即可，不需要过多的考虑表达方式和其内在的逻辑性。其实，这是一个很大的误解。如果本章节的内容表达不清晰，可能会造成读者质疑论文所做出的贡献，甚至是可能否定整篇论文。

那么，撰写相关研究章节的目的是什么呢？相关研究章节起着承上启下的作用。如果在概述章节已经将研究目的表达清楚，那么，接下来就是介绍研究现状了。描述研究现状的时候，可能遇到两种情况。第一种情况是论文所研究的问题已被研究过，有一些或者有很多的相关资料作参考。那么，在这种情况下，相关研究章节只需将其最核心的部分描述清楚即可。另一种情况是，如果本论文开辟了一个新的领域或者问题，目前没有相关的资料可供参考。那么，可以描述一些近似的，但是不同的问题和方法，以表示在这个研究上的空白。有些论文在遇到第二种情况时会进一步分析扩展已有方法来解决目标问题的难度和可行性。这也进一步增强了本研究工作的必要性，因为有些读者可能会有疑问，虽然目标问题是一个新问题，但是如果能够通过简单地，稍稍修改已有方法就能解决问题，那么，就不需要开展研究工作了。

第二个目的，也是更重要的一个目的就是表达论文的独特性。怎样的论文具有独特性呢？首先，提出了全新的问题，并且使用全新的方法解决该问题的论文，具有独特性。这种类型的论文做出的贡献非常大，影响力也非常大。其次，论文提出了一个全新的问题，但是使用的是一种已有的方法，或者整合几种已有的方法，来解决这个新问题。此类论文的贡献也非常大。第三，论文研究



的是一个已有的问题，但是使用的是一种新方法来解决这个问题的。目前来说，绝大多数论文属于这个类别，此类论文做出的贡献常常被称为增量改进贡献。

从上述三个类别可以看出，具有独特性的论文要么在研究的问题上有所创新，要么在问题的解决方法上有所突破。突出前者的创新性相对容易一些，可以直接声明所研究的问题是一个全新的问题。如果能够找到类似的问题，那么，可以比较一下问题定义（后续会介绍如何做比较）。如果是第三类论文，那么就需要强调新方法的独特性，即展示和已有方法相比，新方法具备的一些优势。一个常用的突出某方法优势的写作技巧就是做比较。比较的方法多种多样，下面我们使用例子来展示一些常见的比较方法。

**例一：逐个比较法** 第一种方法是逐个比较法，即将新提出的问题和方法与已有的问题和方法逐一比较。以下是逐一比较的常见写法。

- Zhang proposed a method that utilizes ... for improving ..., while our advanced method primarily focuses on ...
- Unlike an algorithm that improved the performance at a low level, our proposed algorithm optimizes user interfaces.
- Differing from a solution which analyzed the read pattern from the perspective of file systems, our solution explores the pattern from a specific application.
- Our method differs a method at the energy consumption prediction model, in which ... is considered as a key factor in the model.
- A critical observation/behavior that has not been investigated in literature is X. Our solution explores X by establishing a model ...
- Motivated by recognizing the lack of studies in ..., we analyze ...
- X is an important factor in the topic of ..., however, to the best of our knowledge, no method has been proposed to solve X in literature.

**例二：逐类比较法** 第二种方法是逐类比较法。该方法首先罗列了某类问题的方法，然后再将该类问题作为一个整体与论文新提出的方法做比较。以下是一个常见的逐类比较的写法，以供参考。

Zhang proposed METHOD A that was able to ... [1]. Zhang and Jiang improved METHOD B by considering ... [2]. Zhang *et al.* extended SOLUTION C by utilizing ... [3]. METHOD D was developed for ... [4]. APPROACH E was introduced and investigated in ... [4]. SOLUTION F was proposed for solving ... [6]. However, our advanced solution differs from the aforementioned methods at the following aspects. First of all, ...

另外，上述的例子还展示了采用主动句式和被动句式描述相关文献的方法。当以作者名为主语时，如果该文献只有一名作者，则只需给出该作者的姓即可；如果该文献有两名作者，则应同时





给出这两名作者的姓，并用and连接；如果该文献有三名或者三名以上作者时，则应给出第一作者的姓，并紧接et al.。et al.使用斜体（最后一个点表示et al是缩略词，不能省略）。

在讨论已发表的研究文献时，建议使用过去时态。然而，在描述方法内容或者工作原理时，也可用现在时态。因为该方法的内容一般是不变的，过去如何工作，现在还是这样工作。

文献引用（[1]）的位置一般出现在描述该文献的第一句的末尾，或者出现在方法名/作者名之后。两种方法都是常用的方法。当一个句子引用了两篇或者多篇文献时，如果将文献编号全部罗列在句尾，则可能会造成误解或者产生歧义。所以，在这种情况下，建议使用第二种方法。以下是使用第二种方法的一个例子。

**例：**

Method A [1], Approach B [2], and Solution C [3] were proposed for solving ...

在上述的例子中，Method A，Approach B，和Solution C可替换为具体方法的名称。

通常，相关研究章节会出现在概述章节之后。然而，在一些特殊情况下，相关研究章节会被安排在实验结果章节之后，总结章节之前。如果遇到以下其中一种情况，可考虑将相关研究章节安排在靠后位置。

1. 论文在概述章节已对新方法有所介绍。作者认为，文章应直接进入设计与实现章节，以免打断读者的思路。
2. 论文所研究的是一个全新的问题，目前还没有相关方法提出。相关章节的内容并不能很好的突出强调本篇论文独特性的特点。
3. 读者对论文所研究的问题特别熟悉，如果没有阅读相关研究章节，读者也能对当前最新和最有效的方法有所了解，并对论文做出的独特性贡献有所掌握。

## 2.5 基础知识 (Preliminaries)

基础知识章节是一个可选的章节。根据需要，本章节主要用来介绍一些基础知识，以便于读者能够更好的理解设计与实现章节的内容。有些情况下，论文研究的问题比较复杂，需要较大篇幅的文字，甚至是需要使用数学工具来解释。这时可以将问题定义独立设置为一个章节，命名为问题定义（Problem Definition），或者也可以考虑将其设置为基础知识章节中的一个子章节。这样做可以简化设计章节的结构和逻辑。

在论文写作时，一般会假设读者具备了一定的计算机基础知识。但是不应假设读者是某领域的专家。但凡涉及到某领域的特定知识时，这些知识内容是需要详细介绍的。例如，在并行计算领域，以下两个知识点常被设置为独立章节解释其工作原理。（1）MapReduce框架：介绍MapReduce框架的主要目的是为了突出与其他并行计算框架的不同之处。因为shuffling是MapReduce框架特有的



一个处理过程，有些论文会着重讲解shuffling的工作原理。(2) GPU结构。利用GPU的计算能力来增强程序处理性能也是一个并行计算的发展方向。因为GPU工作原理与CPU不同，而且GPU与CPU之间的数据传输有可能成为新算法的性能瓶颈，所以，与GPU相关的论文一般都会详细的介绍GPU的工作原理，以及强调新算法使用了哪些 GPU 的特性，以增强算法性能。以下是一篇有着MapReduce和GPU介绍的论文例子，以供参考。

本例引用自 Wenlu Wang, Ji Zhang, Min-Te Sun, and Wei-Shinn Ku. A Scalable Spatial Skyline Evaluation System Utilizing Parallel Independent Region Groups. The VLDB Journal, 2018.

为了更好的阐述本章节的作用，下面的例子不仅在基础知识章节定义了所研究的问题，还详细介绍了当前对于费马韦伯（Fermat-Weber）问题研究的现状。费马韦伯问题是一个世界著名的难题。在数学界，研究人员提出并证实了很多成果。有些成果已经应用于人们的生活中。在本例中，该论文所提出的新方法需要用到费马韦伯问题的解法来寻找在多维空间中的一个最优位置。因为计算机领域的研究人员并不熟悉费马韦伯方法的内部细节，所以，该论文使用了一个子章节介绍这个问题的研究现状和目前的研究结论。

被人们熟知的费马问题是，在空间中任意给定三点，并在空间中寻找一个最优点，使得从这个最优点到给定的三个点的距离和最小。费马韦伯进一步扩展了这个问题。它将给定的点数扩展为 $n$ 个，并且为每个给定的点设置了一个权值，按照加权的距离和最小来寻找最优点。根据目前的研究现状可知，当给定点共线时，最优点的位置可在线性的时间内找到，找到的点是精确的最优点。但是在三个以上给定点且不共线的情况下，以目前的研究水平还无法找到精确的最优点。但是，可以以一种迭代的方式无限逼近这个最优点。这个迭代的方法从任意一个（不与给定点重叠的）点 $q_0$ 开始，不断的调用迭代函数 $q_{i+1} = f(q_i)$ 。由该迭代函数单调收敛（Monotonic Convergence）的特性可知， $q_{i+1}$ 比 $q_i$ 更接近于这个最优点。理论上讲，这个最优点应该无限接近于 $\lim_{i \rightarrow \infty} q_i = f^i(q_0)$ 。若对本例的内容感兴趣，可参考该论文章节 2.3。

本例引用自 Ji Zhang, Wei-Shinn Ku, Min-Te Sun, Xiao Qin, and Hua Lu. Multi-Criteria Optimal Location Query with Overlapping Voronoi Diagrams. In Proceedings of the 17th International Conference on Extending Database Technology (EDBT), Athens, Greece, 2014. 本例的全文可在如下网址处免费获取。  
<https://openproceedings.org/2014/conf/edbt/ZhangKSQL14.pdf>

## 2.6 新模型 (Model), 或者新概念 (Concept)

如果论文提出了一个新的模型或者概念，那么，建议将其在一个独立章节中表述。这样做的好处是突出了这个新模型或者概念对于这项研究的贡献，也增加了论文的影响力。

新模型，新概念是与设计/实现不同的。一个具体的设计与实现往往建立于某一假设或者某一运行环境的基础之上。所以，这个设计与实现的影响力也受到这个假设或者环境的限制。然而，新模型和新概念是在更加抽象的层面中的一种创新。它基于更少的假设，但可适应于更广的应用环境之中。

满足以下几个特点的论文可考虑独立设置新模型/新概念章节。



1. 论文能够建立一个新的概念或者模型，并利用该概念和模型描述一个抽象的问题。这个新的概念或者模型内容充实，新颖，能自成体系。
2. 论文所研究的问题恰好是这个新概念/新模型的一个具体应用。
3. 提出新的概念或者模型能帮助诠释论文所提出的新方法。例如，可提供正确性的证明，或定性地分析算法复杂度等。
4. 新的概念或者模型有较广的应用面。除了应用于论文所研究的问题外，还可应用于许多其他的问题。

为了更好的展示新模型/新概念章节的写作方式，以及该章节与设计实现章节的区别，本文用如下例子详细介绍该论文第四章中建立的MOVD模型。

本例引用自 Ji Zhang, Wei-Shinn Ku, Min-Te Sun, Xiao Qin, and Hua Lu. Multi-Criteria Optimal Location Query with Overlapping Voronoi Diagrams. In Proceedings of the 17th International Conference on Extending Database Technology (EDBT), Athens, Greece, 2014. 全文可在如下网址处免费获取。  
<https://openproceedings.org/2014/conf/edbt/ZhangKSQL14.pdf>

### 例:

该论文所研究的问题是在整个搜索空间中寻找一个最优点，而论文中建立的MOVD模型能高效的帮助寻找该最优点。具体的讲，MOVD模型定义了一个操作数MOVD，和一个二元操作符 $\oplus$ 。 $\oplus$ 可将两个MOVD叠加起来，生成另一个MOVD。该论文提出的新方法需要将多个MOVD叠加在一起，以便于快速找出最优点。

每一个MOVD是一个d维空间的划分，它按照某一种规范(Criterion)，将整个搜索空间划分成多个子空间（MOVD是可以支持d维空间的，但论文中仅以二维空间作为例子说明）。最简单的MOVD，或者称为原子MOVD，是一个Voronoi Diagram。然后，将二个或者多个 MOVD叠加之后，生成的还是一个MOVD。这样就建立起了一个新的代数结构(Algebraic Structure)。

那么，该论文为什么需要提出这个MOVD的代数结构呢？在撰写论文之初，作者并未意识到这是个代数结构，也没有意识到这篇论文需要建立一个代数结构。然而，在对该论文进行不断完善的过程中，作者逐渐发现了一些问题，找到了相应的解决方法，提出了更多的关于MOVD以及叠加操作符 $\oplus$ 的属性，最后，作者将所有的属性汇总在一起，形成了这个代数结构。

在这个完善过程中，作者提出了以下问题

1. 如何建立一个MOVD？

答案是：MOVD都是由原子MOVD（或者Voronoi Diagram）叠加而成的。

2. 给定一组原子MOVD，叠加后的结果一定存在吗？是唯一的吗？

由这个问题，作者得到了多个结论。1）叠加后的MOVD一定存在，并且唯一。其存在性可有MOVD的定义得到。唯一性是由以下几个属性共同支撑的。2）可交换性和可结合性。即，左操作数和右操作数互换不改变叠加操作的结果。



### 3. 叠加两个MOVD，一定会生成另一个MOVD吗？

这个问题其实是在质疑，如果在某种情况下，叠加操作生成的不是一个MOVD，那么，叠加操作就无法继续叠加下一个MOVD了。因为本论文定义的叠加操作仅能应用于叠加两个MOVD，不能应用于其他操作数类型。幸运的是，论文最后得到结论，叠加操作是在MOVD集合中封闭的，即，叠加两个MOVD，一定生成一个MOVD。

### 4. 每个原子的MOVD都覆盖整个搜索空间，那么，叠加之后生成的MOVD也覆盖整个搜索空间吗？

答案是肯定的。以上四项其实是为了证明论文提出的新算法的正确性。因为，正是由于MOVD覆盖了整个搜索空间，搜索整个空间和搜索MOVD是等效的。从而，新方法将问题转化为在一个MOVD中寻找一个最优点的问题。

得到以上结论后，这个代数结构的雏形就已经建立起来了。此后，作者还增加了一些辅助的属性，比如MOVD空间、在一个给定的MOVD空间中计算MOVD的个数等。这个结论有助于分析论文提出的新算法的时间复杂度。

## 2.7 设计与实现 (Design and Implementation)

本章节是整篇文章最核心的部分。前面所讨论的章节，例如概述(Introduction)，相关研究(Related Work)，新模型/新概念(Model/Concept)等，都是为了引出新方法的设计与实现和突出新方法的新颖性和独特性的。在本章我们着重介绍如何构思和表述设计和实现章节。每篇论文设计与实现章节的内容各不相同，下面是三种常见于各种论文中的方法，即自顶向下法，由浅入深法和抛砖引玉法。

### 2.7.1 自顶而下(Top-Down)法

自顶而下的方法是一种常见的方法。该方法从解决方案的整体出发，将各个部分以模块的形式组织在一次。该方法首先给读者一个宏观的介绍，然后，再逐步细化各个部分的内容。基于此方法的论文，一般以解决方案框架(Framework)或者整体介绍(Overview)入手，将其独立设置成一个子章节，或者出现在第一个子章节之前。然后，其他各个部分自成一个子章节，或者以某种逻辑将几个部分安排在一个子章节中。

**例：**

本例给出的这篇论文在第二章使用了自顶而下的方法。该论文将相关研究章节安排在实验章节之后，因为Skyline Query是在该领域内较为熟知的研究问题。如图 1所示，作者在第一个子章节(2.1)中，首先介绍了新方法的输入与输出，以及方法内部的三个步骤，并且简洁的阐述了每一个步骤所使用的概念。换句话讲，这个子章节讲述了新方法的整体流程。然后在第二个子章节(2.2)着重介绍新方法的第一个步骤，即寻找MBR集合中的Skyline元素；在第三个子章节(2.3)中介绍了第二个步骤，即多个依赖组的生成方法。在图 1中，明确标识了后续描述算法在新方法中的位置，



所以，依据此图，读者能够清晰的看到整个设计章节的组织结构。因为第三步骤非常简单，所以，在第二章的最后，作者仅用了一小自然段阐述第三个步骤的内容，并再次快速总结了整个新方法的处理流程。

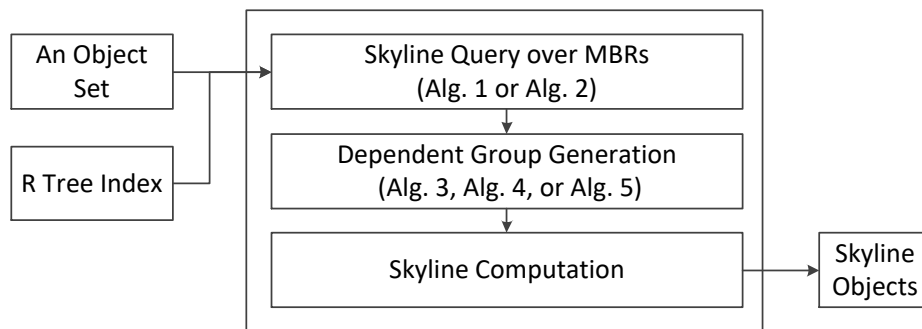


图 1: 自顶而下法。

本例引用自 Ji Zhang, Wenlu Wang, Xunfei Jiang, Wei-Shinn Ku, and Hua Lu. An MBR-Oriented Approach for Efficient Skyline Query Processing. In Proceedings of the 35th IEEE International Conference on Data Engineering (ICDE), Macau SAR, China, April 2019.

### 2.7.2 由浅入深法

由浅入深的方法遵循从易到难的思路，即先解决一个相对简单的问题，然后逐步增加问题的复杂度和难度，直到完全解决所研究的问题为止。有时，某些论文所研究的问题比较复杂，或者所提出的新方法比较复杂，如果直接给出完整的解法，读者可能会比较难以接受。由浅入深法非常适用于这种情况。论文可以从一个较为容易理解的场景，或者贴近生活的用例入手，逐步增加难度，并且使用例子辅助读者理解。

以下论文在设计章节（5.3）中，使用了由浅入深法证明一个重要的概念Pruning Region。整个章节的核心就是提出这个概念（Theorem 2）和证明由这个概念得出的一个规则。此例仅仅是为了介绍由浅入深的用法。如果对其证明细节感兴趣，可参考论文原文。

### 2.7.3 抛砖引玉法

抛砖引玉法的思路是以问题为驱动力：先抛出一个问题，再解决这个问题。这个问题可能存在于已经发表的方法中，也可能是一个新问题。而抛出这个问题是为了引出论文将要提出的新方法。

以下用例引用的论文使用了抛砖引玉法来提出两个数据划分的方法。这两种方法严格控制每个划分中数据量的大小，以加快数据分割后的并行计算的处理速度。读者如对其论述过程感兴趣，可参考论文原文。





## 例

经作者观察，使用经典算法MaxSum和MaxDist划分数据的结果并不令人满意。这两个方法都同时进行数据划分和数据过滤。在处理过程中，某些区域的数据被过滤的较多，而另一些区域被过滤的较少。这就造成了在数据过滤切分之后数据分布不均匀的情况。然而，在后续的并行计算中每个分区被独立处理，整个并行计算的用时取决于处理拥有最多数据的分区的时间。为了改进数据过滤和划分的过程，并提出新的方法，该论文按照如下逻辑组织了这个子章节内容。

首先，论文做了一个简单的实验(Preliminary Experiments)，使用一个表格展示了经MaxSum和MaxDist处理之后，各分区包含数据的个数。这个实验结果证明了作者的一个观察，即使用MaxSum和MaxDist会造成数据分布不均匀的问题。然后，论文阐述了一个基本的并行计算的理论观点，整个并行计算的用时由最慢的那个处理程序决定。所以，论文指出需要一个更优的数据划分的方法来改进这个并行计算的过程。

在准备提出新方法之前，经作者的分析，提出新方法最根本的目的其实不是解决数据分布不均的问题。其根本问题是那个最“胖”分区包含了太多数据。所以，第一个新方法着重关注最“胖”的那个分区，将数据过滤重点放在这个最“胖”的分区上。但是，当前最“胖”的分区并不一定是最后那个最“胖”的分区。提前将重心放在当前最“胖”的分区并不一定能得到最好的结果。所以，论文还提出了第二个新方法，将数据过滤的重点按照“胖”“瘦”比例分配到各个分区上。

值得注意的是，上述实验只是为了暴露MaxSum和MaxDist存在问题。但实验结果不能证明新方法优于MaxSum和MaxDist。这是需要在后续实验结果章节中给出定量的分析，以作为前后呼应。

本例引用自 Ji Zhang, Xunfei Jiang, Wei-Shinn Ku, and Xiao Qin. Efficient parallel skyline evaluation using MapReduce. IEEE Transactions on Parallel and Distributed Systems (IEEE TPDS), Vol. 27, Issue. 7, pp. 1996-2009, 2016.

## 2.8 分析 (Analysis)

分析章节是一个可选章节；它一般出现在设计章节之后，是对设计章节的补充和延展。因研究问题的不同，分析章节的内容也不尽相同。有的文章在分析章节中着眼于设计本身，进行一些理论的定性分析。而另一些文章的分析章节则将重点放在新方法的应用和扩展上。

本节我们仅给出一些撰写分析章节的思路，以供读者参考。

1. 算法复杂度分析。算法复杂度分析是一种常见的分析内容。算法复杂度分析可分为时间复杂度分析和空间复杂度分析两种。一般情况下，假设算法输入数据量为 $n$ ，那么，分析的目的是找出一个与 $n$ 相关的复杂度函数 $f(n)$ 。这个复杂度可能还会分为最优情况下的复杂度，平均复杂度，和最坏情况下的复杂度。

以上内容都是常规分析，我们可以尝试再进一步。如果输入数据满足一定条件，那么，是否能够得到更加有意义的分析结果呢？例如，假如输入数据遵循均匀分布，那么，是否能得到一



个更加精确的算法时间复杂度或者空间复杂度呢？这个分析背后的意义在于，假如有一种应用场景，其输入数据是均匀分布的，那么，上述的分析结果就可以直接应用到这个场景中。当然，不同的应用场景，输入数据的特点是不同的，这需要根据研究问题的特性相应变化。值得指出的是，如果按照这个思路，假设输入数据满足一个具体的概率分布函数/概率密度函数，那么，概率和随机过程的知识是一个很好的可利用的分析工具。

2. 对于特殊情况处理的分析。例如，对于一个网络设计方法，如果网络出现错误，丢了一个包，这个新方法多快能恢复正常？当处于某情况下，丢包率是多少？这个丢包率与哪些因素相关？
3. 与其他方法作比较。当新方法叙述完成之后，将其与老方法做个比较也是一种常见的分析方法。在相关研究章节，可能因为需要比较的方法太多，不可能和每个相关的方法做详细比较。那么，分析章节就是一个非常适合与类似方法做比较的地方。

## 2.9 实验结果 (Experimental Validation)

### 2.9.1 实验环境搭建

绝大多数计算机研究领域是需要由实验结果来验证以支持新方法的有效性的。因为计算机实验具备可重复性，即在给定的实验环境下，无论何时何地，计算机运行的结果都是相同的。所以，实验结果章节需要详细地给出实验环境，实验数据以及操作过程，以便于其他研究人员重现实验结果。

基于以上原因，实验章节的第一部分一般是详细介绍实验环境搭建，数据准备和实验的操作过程。这其中包括：

1. 实验环境（软硬件环境）搭建。硬件包括：CPU型号，内存大小，硬盘型号，以及网络配置等。软件一般包括操作系统，实现新方法所使用的软件工具等。
2. 实验数据准备。实验数据需要给出数据的来源，数据大小(The size of datasets)，和数据特点（例如：数据分布）等。如果数据经过了预处理，则需要详细说明预处理的过程和预处理的必要性。真实的数据(Real-world datasets)需要给出数据来源；合成的数据(Synthetic datasets)需要给出生成算法或者使用的生成工具。
3. 新方法的实现细节。例如，使用的编程语言以及重要的第三方代码库。
4. 比较方法实现的细节。如果是自行实现的，需要给出使用的编程语言和重要的第三方代码库。如果是下载的，则需要给出来源。
5. 简洁的介绍一下实验参数，以及其默认取值。
6. 介绍一下实验是如何组织开展的，以及后续实验章节的安排。

在实验部分，一个常见的问题是使用真实数据还是合成数据？一般来讲，这两种数据各有用途。通常使用真实数据做出的结果更具有说服力。毕竟计算机是一门应用学科，它更侧重于新方法



在实际应用中的有效性。然而，有时使用真实数据很难完全展示新方法的特点，或者获取真实数据存在一定难度。所以，在这类情况下，使用合成数据是一个较好的选择。总而言之，在真实数据与合成数据的取舍当中，我们更应该从实验目的以及实验结果的展示效果着手，选择实验数据。

### 2.9.2 实验内容设计

在将实验的基本信息介绍清楚之后，接下来就是描述具体的实验结果了。实验结果一般会围绕几个不同的侧重点展开描述。具体如何操作，应视研究内容而定。然后，它们之间有一些相通性。以下列举一些常见的实验内容，以供参考。

1. 测试输入数据的变化对新方法的影响（例如，改变数据大小，维度，分布等）。
2. 测试运行环境的变化对新方法的影响（例如，改变CPU处理能力，I/O速率，CPU个数等）。
3. 测试新方法中某数据结构变量对新方法的影响（例如，将二叉树的算法修改为三叉树算法等）。
4. 测试策略的变化对新方法的影响（例如，将最老优先替换策略修改为最少使用优先替换策略等）。
5. 比较分析章节给出的分析与实验运行的结果（例如，假如分析章节给出的分析结果是线性的时间复杂度，那么，这个分析结果应在实验结果中得到验证）。

### 2.9.3 实验制图

为了将实验结果有效合理的展示出来，实验作图工具必不可少。笔者常常使用gnuplot工具绘制柱状图，折线图，和三维折线图。gnuplot是一个免费的作图工具。gnuplot作图过程由命令驱动，能画出多种多样的科学用图。本文着重介绍使用gnuplot工具生成柱状图、折线图和三维折线图。在这里查看gnuplot的相关信息。

以下五个例子均生成eps文件格式的图片。eps图片是矢量图片，在论文中，可以任意放大和缩小而并不损失图片的清晰度。正是因为这个原因，论文中应尽量使用矢量图。

#### 柱状图

柱状图常用于多方法之间比较某一个具体的值。这个值可能是运行时间、开销、距离等指标。如下例所示，该图比较了四种方法的执行时间。执行时间的单位为秒。该图通过由柱状在Y轴的高度来体现方法的优劣。有时，柱状的高度差别很小，Y轴从原点开始，可能无法很好的显示方法之间的差异。所以，此时可只显示Y轴的某个区间（例如：[50-100]）。

在使用gnuplot绘制实验图时，笔者将gnuplot的绘制指令和绘制数据分别存放在两个文件中。例如，可将这两个文件命名为bar.dem和bar.data。在存放这两个文件的目录中，执行"gnuplot bar.dem"，则gnuplot会自动地生成图片文件bar.eps。这两个文件的内容如下。





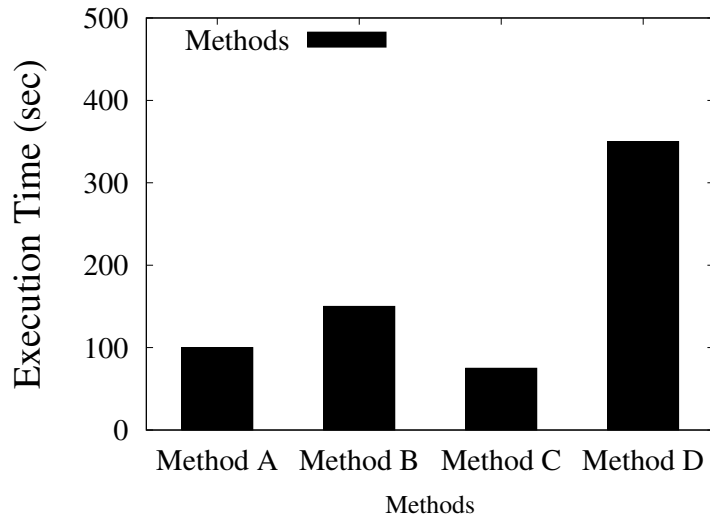


图 2: 柱状图。

bar.data文件的前两行设置了文件输出的格式是eps格式，字体是Times-Roman，28号字。输出图片的文件名为bar.eps。set xlabel和set ylabel分别设置X轴和Y轴的标签内容，以及使用的字体和字号。set key设置了caption放在图片的左侧顶部。set xrange和set yrange设置X轴与Y轴的取值范围。如果只显示X轴和Y轴的某一值域时，可在这个命令中设置。第一个字符井号#表示当前行被注释掉了。set boxwidth设置柱状的宽度。set style则是设置柱状着色的样式。plot命令是绘制图片的命令。它会根据当前已设置的状态，绘制出图片。第一个参数"bar.data"指明了数据存放在这个文件中，using指明了使用数据文件中第一栏和第三栏的数据绘制图片。xtic(2)则指明了使用第二栏数据作为柱状的名字。title设置了在caption中的名字。with boxes表示这是绘制的柱状图。

#### 源代码示例

```
set term postscript enhanced eps font "Times-Roman, 28"
set output "bar.eps"
set ylabel "Execution Time (sec)" font ",36"
set xlabel "Methods" font ",24"
set key left top
#set xrange [0:1200]
set yrange [0:500]
set boxwidth 0.5
set style fill solid
plot "bar.data" using 1:3:xtic(2) title "Methods" with boxes
```

相对应的数据文件bar.data的内容如下。

#### 源代码示例

```
0 "Method A" 100
```



```
1 "Method B" 150
2 "Method C" 75
3 "Method D" 350
```

## Histogram图

当使用柱状图展示某数据的总和时，可使用Histogram图将各部分的数据也展示出来。

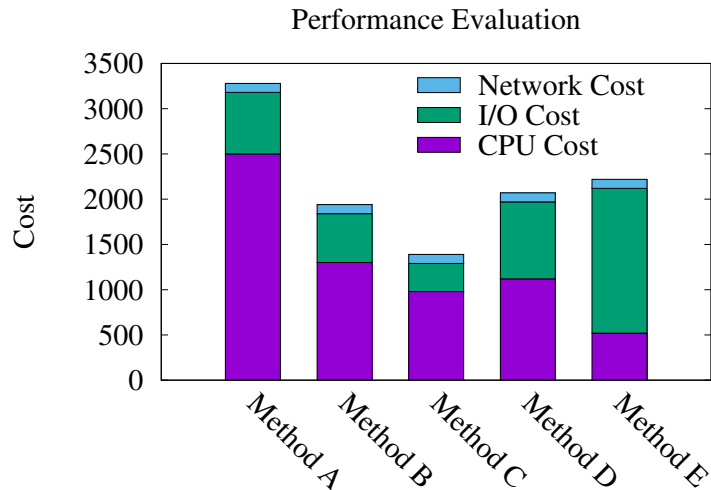


图 3: Histogram图（例一）。

本例使用的是`set style histogram rowstacked`。gnuplot同时还支持`columnstacked`。`rowstacked`指的是，在数据文件中，按行叠加，即在某行内的数据叠加在一个柱状上；而`columnstacked`指的是，按列叠加，即在某列内的数据叠加在一个柱状上。

### 源代码示例

```
set term postscript enhanced eps color font "Times-Roman, 28"
set output "bar-2.eps"
set boxwidth 0.6 absolute
set style fill solid 1.00 border lt -1
set key right top vertical Left reverse noenhanced autotitle columnhead nobox
set key invert samplen 2 spacing 1 width 0 height 0
set style increment default
set style histogram rowstacked title textcolor lt -1
set style data histograms
set xtics border in scale 0,0 nomirror rotate by -45 autojustify
set ylabel "Cost"
```



```
set title "Performance Evaluation"
plot "bar-2.data" using 2:xtic(1), for [i=3:4] "" using i
```

其数据文件内容如下。

#### 源代码示例

```
Cost "CPU Cost" "I/O Cost" "Network Cost" Total
"Method A" 2500 680 100 3280
"Method B" 1300 540 100 1940
"Method C" 980 310 100 1390
"Method D" 1120 850 100 2070
"Method E" 520 1600 100 2220
```

另一种Histogram图常用于展示某数据的平均值。与此同时，该图还可以展示最大值与最小值。如下图所示。

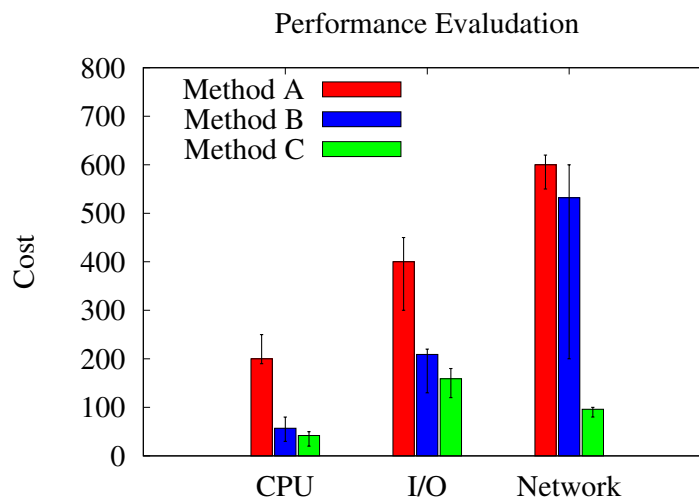


图 4: Histogram图（例二）。

在本例中，使用了set style histogram errorbars。其他的设置与上述例子类似。

#### 源代码示例

```
set term postscript enhanced eps color font "Times-Roman, 28"
set output "bar-3.eps"
set title "Performance Evaluation"
set boxwidth 0.9 relative
set style fill solid 1.00 border 0
set style histogram errorbars gap 3 lw 2
set style data histograms
```



```

set yrange [0:800]
set ylabel "Cost"
set bars 0.5
set key top left
plot "bar-3.data"
using 2:3:4:xtic(1) ti "Method A" linecolor rgb "red", \
"" using 5:6:7 ti "Method B" lt 1 lc rgb "blue", \
"" using 8:9:10 ti "Method C" lt 1 lc rgb "green"

```

其数据文件内容如下。在该文件中，第2-4栏，第5-7栏和第7-9栏分别包含了一组平均值、最大值和最小值。

#### 源代码示例

```

"CPU" 200 250 190 57 80 30 42 50 20
"I/O" 400 450 300 209 220 130 159 180 120
"Network" 600 620 550 532 600 200 96 100 80

```

## 折线图

折线图是最为常见的一种用于展示实验结果的图型。因为折线图不仅能显示多个方法之间的差距，而且还可以展示随着某项指标的变化，各个方法的趋势和走向。为了展示gnuplot的灵活性，下面的例子使用了四种颜色的曲线表示四种方法的执行时间，并且这四种曲线还使用了不同的样式以增强区别，以便于使用黑白打印机的读者能够清晰辨认。黑白图片和彩色图片均能在论文中使用。而且越来越多的论文采用了彩色图片，因为使用彩色能更好的突出图片需要强调的内容。

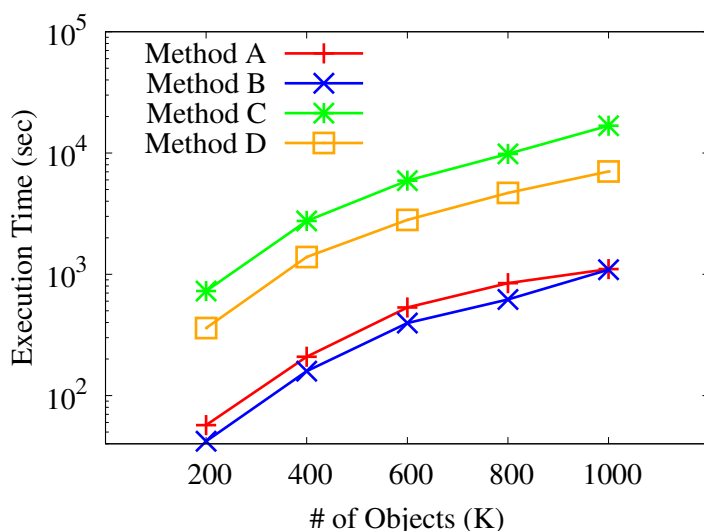


图 5: 折线图。

与绘制柱状图类似，本例也将绘图命令和绘图数据分别存放在plot.dem和plot.data文件中。在指令文件plot.dem中，首先设置了输出图片的格式为eps，图片的全局字体为Times-Roman，28号



字。输出文件为plot.eps。然后，分别设置了X轴和Y轴的标签，caption放置在图片的左上角，设置X轴和Y轴的取值范围。这个例子的一个不同之处在于，由于Y轴的取值范围非常大，这会导致曲线的某些取值点非常低，非常接近于X轴，致使读者无法分辨该点的取值。因此，在此例中，Y轴采用了对数显示的方法，即10, 100, 1000, 10000等距的显示在Y轴上。set logscale y 10指定了Y轴以10为底数的对数的显示方式。set format Y "10^{%T}"则设置了Y轴上各个标记点的值的显示格式。最后，plot命令绘制整个折线图。其中，每一行表示绘制某一方法的样式。例如，第一行绘制Method A。Method A的数据出现在plot.data文件中的第一栏和第二栏，名字为Method A，该折线需要标记出取值点的位置（with linespoints），并且设置折线的宽度为5，使用红色绘制。具体的参数设置方法请参考gnuplot手册。

#### 源代码示例

```
set term postscript enhanced eps color font "Times-Roman, 28"
set output "plot.eps"
set ylabel "Execution Time (sec)" font ",28"
set xlabel "# of Objects (K)" font ",28"
set key left top
set xrange [0:1200]
set yrange [40:100000]
set pointsize 3
set logscale y 10
set format y "10^{%T}"
set xtics (200, 400, 600, 800, 1000)
plot "plot.data" using 1:2 title "Method A" with linespoints lc rgb "red" lw 5, \
"plot.data" using 1:3 title "Method B" with linespoints lc rgb "blue" lw 5, \
"plot.data" using 1:4 title "Method C" with linespoints lc rgb "green" lw 5, \
"plot.data" using 1:5 title "Method D" with linespoints lc rgb "orange" lw 5
```

相对应的数据文件plot.data的内容如下。第一栏为各方法在X轴上的取值，其余四栏则是各个方法在Y轴上的取值。

#### 源代码示例

```
200 57 42 727 360
400 209 159 2749 1388
600 532 396 5919 2809
800 845 620 9830 4688
1000 1104 1091 16774 7041
```

## 三维折线图

在折线图的基础上，可使用三维折线图来展示当有两个参数同时变化时的各个方法的趋势和走向。与绘制折线图不同的是，一个数据文件仅包含一个方法的数据。换句话说，当比较两个方法时，绘制三维折线图需要使用一个指令文件和两个数据文件，以此类推。在本例中，它们是3d.dem，



3d-1.data和3d-2.data。

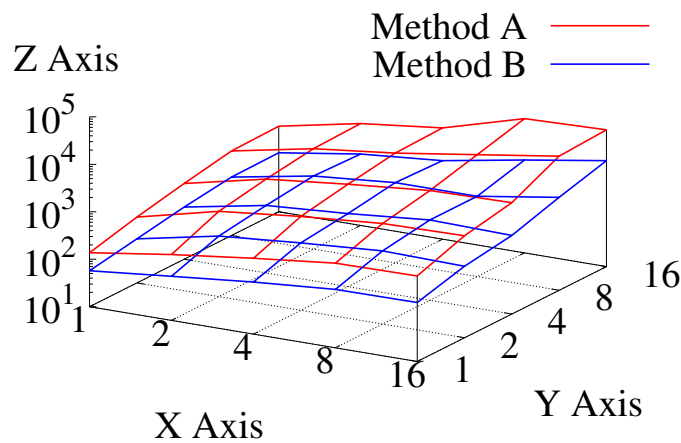


图 6: 三维折线图。

本例的大部分指令与上述例子类似，X轴、Y轴和Z轴分别由两条命令设置标签、字体、字号和位置。offset character可在图片中设置标签文字的位置，norotate和rotate by可用于旋转标签的文字。Z轴使用对数方式展示。在最后，splot命令用来绘制三维折线图，每一行表示一种方法的曲线样式。

#### 源代码示例

```
set term postscript enhanced color eps font "Times-Roman, 36"
set output "3d.eps"
set samples 20
set isosamples 75,75
set ticslevel 0
set view 60, 30, 1, 1
set clabel "%8.2f"
set key right top
set xlabel "X Axis"
set xlabel offset character 0, -1.8, 0 norotate
set ylabel "Y Axis"
set ylabel offset character 0, -2, 0 font "" textcolor lt -1 rotate by -270
set zlabel "Z Axis"
set zlabel offset character 6, 4, 0 font "" textcolor lt -1 norotate
set zrange [10:100000]
set logscale z
set logscale xy 2
set format z "10^{%T}"
```



```
set grid
splot "3d-1.data" using 2:1:3 title "Method A" with lines lc rgb "red" lw 3, \
"3d-2.data" using 2:1:3 title "Method B" with lines lc rgb "blue" lw 3
```

3d-1.data的数据为:

#### 源代码示例

```
1 1 137
1 2 244
1 4 396
1 8 606
1 16 634

2 1 244
2 2 628
2 4 950
2 8 1311
2 16 1408

4 1 396
4 2 950
4 4 1438
4 8 2060
4 16 2215

8 1 606
8 2 1311
8 4 2060
8 8 3712
8 16 6724

16 1 634
16 2 1408
16 4 2215
16 8 6724
16 16 7593
```

3d-2.data的数据为:



### 源代码示例

```
1 1 57
1 2 85
1 4 125
1 8 170
1 16 175

2 1 85
2 2 189
2 4 263
2 8 348
2 16 321

4 1 125
4 2 263
4 4 345
4 8 492
4 16 452

8 1 170
8 2 348
8 4 492
8 8 492
8 16 907

16 1 175
16 2 321
16 4 452
16 8 907
16 16 1710
```

## 2.10 总结及未来计划 (Conclusion and Future Work)

本章节总结论文，以及提出未来该课题可扩展的方向。本章节应简短地总结文章的贡献以及实验结果。未来计划是可选部分。但如果需要的话，应基于本篇论文成果的基础上，提出扩展方向。





## 2.11 致谢 (Acknowledgement)

致谢部分主要是对为本研究做出贡献的人或者组织表达感谢。此处，对本研究的贡献可包括，但不限于，（1）参与过本研究的讨论；（2）对研究问题曾给出有帮助的建议；（3）参与或改进文章写作；（4）对本研究给予资金或者其它资助等。

## 2.12 参考文献列表 (Bibliography)

参考文献章节列出了所有论文中引用了的文献。这些文献需与研究问题相关。大多数会议或者期刊会以第一作者的姓氏顺序排列文献。



## 3 常见问题 (FAQ)

### 3.1 如何寻找一个合适的研究问题或者研究方向？

无论是对初级研究人员还是资深研究人员，寻找一个合适的研究问题和方向是从事科学研究的一个非常关键和重要的环节。初级研究人员常常为找不到合适的研究问题而苦恼。以下笔者将分享一些常用的寻找研究问题和研究方向的经验，以供参考。

1. 对于（博士/硕士）研究生而言，最直接而有效的方法是询问导师或者师兄师姐。一般来说，在开始进行学习和研究之前，导师会为每一名研究生制定一套完整的，具体的科研计划。该计划一般包括科研项目的目标，研究内容，研究手段等信息。这个计划的制订可能由导师决定（毕竟有些科研项目是有科研课题经费资助的），或者由导师和研究生讨论共同确定。
2. 如果研究内容比较自由，允许视情况而定的话，研究人员可使用如下方法尝试寻找最佳的研究方向。

方法一：在确定研究领域后，可通过查找和翻阅近年来在顶级会议或者期刊上发表的文章来寻找新方向。这些文章都是高质量的文章。它们思路清晰，能够将某一具体问题解析得非常透彻。通过阅读这些文章，还可以对这些文章研究的问题有一个整体的把握（包括问题背景，研究现状，以及解决方法中存在的缺陷等）。以此类文章为参考是一个值得尝试的方法。本文附录中罗列了各个方向的会议和期刊列表，以供查阅参考。

方法二：在文献搜索服务中搜索关键字，查找相关热门文章。一般来说，被引用较多的文章会排在靠前的位置。这些文章往往不是最新的（最新文章引用往往不会很高），但是它们非常具有启发性。因为从引用它们的文章中，读者可以仔细观察其他文章是如何扩展该篇高引用文章的。然后，读者可以尝试着从一个新的方向/角度继续研究和完善该问题。在第 3.2 章节可找到推荐使用的文献搜索服务。

方法三：积极参加国际会议。在会议中不仅能和其他研究人员相互交流心得、激发灵感，而且还能了解领域内最新的研究成果和研究方向。会议一般会邀请一名或者多名资深研究人员给keynote talk。这个keynote talk一般是介绍和推广该资深研究人员非常看好的研究方向或者交流一些重要的研究经验。

方法四：关注一些ACM或者IEEE出版的刊物。ACM和IEEE会定期的邀请资深研究人员撰写评论文章或者概述性的文章。这些文章通常会探讨某一学术问题或者某一领域的发展现状以及对未来的展望。这些展望内容是经过深思熟虑的，甚至是经历过一些研究探讨或者实验验证的。如果能从中发现一些好的问题难点，是值得花时间去探索和研究的。

### 3.2 如何搜索国际文献信息？

计算机领域较为常用的文献搜索服务是谷歌学术(Google Scholar) 和dblp。谷歌学术可以查找各个学科的信息（包括计算机领域），而dblp则专注于计算机领域文献信息收集和检索。



LittleWaterdrop也提供文献搜索服务。该服务能够按照作者名或者文章题目中包含的关键词查找文献，并在结果页面中提供文献官方页面的链接。这些文献及作者信息都来源于dblp提供的数据库。

### 3.3 如何阅读一篇论文？

因目的不同，阅读论文的方法也不尽相同。以下是三种较为常用的阅读方法，分别应用于三种不同的场景。

1. 快速阅读法。使用快速阅读方法，读者可在几分钟之内掌握一篇论文的要点。当然，该方法不会要求读者进行逐行逐字阅读。一般此种方法常用于在海量文献中寻找读者感兴趣的文章。当开始新的一项课题研究或者寻找一项新的研究方向的时候，研究人员可能会收集大量的论文，并逐篇快速阅读。由于研究人员可能只是对所读文章研究的问题感兴趣，或者对文章中提出的方法感兴趣。此时，研究人员只需阅读文章摘要即可。

另外，快速阅读方法是否能成功应用，还取决于所读文章是否具有高质量的摘要，高质量的摘要能够清晰的描述该论文研究的问题，提出的新方法，以及该方法的有效性。抓住这些要点实际上就是抓住了文章的核心内容。

2. 详细阅读法。当研究人员确定某些文章值得更深入的研究时，可对其展开详细阅读。详细阅读不仅要求读者逐行阅读，而且还需要理解每个图表，例子，算法等。并且在适当的时候，还需要查找相关的文献，以印证，对比所读文章的内容。
3. 跳跃阅读法。这种方法常常应用于文章评审中。当审稿人拿到一篇待审文章时，首先需要确定的是该文章是否在其熟悉的领域内，审稿人是否具备足够的知识、能力与经验来评审这篇文章。所以，通常审稿人会先阅读文章的题目和摘要。在确认继续进行评审后，评审人可直接跳跃至概述章节的中部或尾部，以了解文章的主要贡献。如若评审人非常熟悉这个领域，那么解释问题重要性的内容可以快速跳过。在确定文章主要贡献和创新性后，审稿人可以直接阅读文章的设计章节；也可以先阅读现有方法存在的问题后，再跳到主体部分。

注意，以上描述的只是一种审稿阅读法。不同审稿人会有不同的阅读策略，但在审稿过程中，审稿人大多会按照文章各部分的重要程度和逻辑顺序排列优先级，并按照优先级顺序进行阅读评审。

跳跃阅读方法适用于具有较为固定格式的论文，它们能使审稿人快速找到相应的内容。

### 3.4 会议与期刊的区别

计算机领域的学术会议一般指Conference。研讨会Symposium有时也会被粗略的认为是一种会议。绝大多数会议一年举办一次，少数会议两年举办一次。

会议论文投稿需要注意几个重要时间点：投稿截止时间，审稿意见通知时间，和最终版本提交时间。有些会议设置了两个截止时间：摘要截止日期(Abstract Deadline)和论文提交截止日期(Submission



Deadline)。这些会议要求作者在摘要截止日期之前注册稿件并提交一份论文的摘要，然后，在论文截止日期之前完成论文全文的提交。这种做法是为了让会议主席先对稿件的方向，内容，和数量有一个大致的了解，然后根据这些数据邀请相关专家评审稿件。

一般来说，会议的审稿时间为投稿截至之后的1-2月。大多数会议只有一轮审稿。会议主席通常会为每个稿件邀请至少三名专家进行评审。每个评审意见大致可分为五档：Accept, Week Accept, Borderline, Week Reject, 和Reject。会议主席会根据稿件初审结果来做决定。当所有稿件的评审结果决定后，会议主席会给每篇论文的作者（有可能只是第一作者，或者提交稿件的作者）发送评审结果和审稿意见。这一流程一般会在公示的审稿意见通知时间(Notification)之前完成。因为审稿人可能会对稿件做出误判（虽然概率很小），所以，目前有些会议在审稿结束之后新增了一个Rebuttal的过程。在这个过程中，作者有机会给审稿人反馈意见。当审稿意见返回给作者后，如果作者认为审稿人做出了错误的判断，或者审稿人错误的理解了论文的内容，作者可以给会议主席和审稿人反馈。最后，如果论文被会议录用，会议主席会要求作者根据审稿意见修改稿件，并准备最终的论文版本。这个版本被称为Camera Ready版本。提交最终版本的截止时间常被称为最终版本提交时间(Camera Ready Deadline)。

绝大多数会议要求每一篇录用的文章需要至少有一名作者参加会议，并在会议上讲解该文章的内容。

计算机领域的学术期刊一般指Transactions或者Journal。在别的学术领域，一般只有期刊被认为是正式发表的作品，会议论文只是一个简稿，或者手稿，仅作交流之用。然而，计算机领域与其他领域是不同的。因为计算机科学技术发展得非常快，而期刊审稿的周期较长，发表在期刊上的论文内容可能不能跟上最新技术的发展速度。所以，逐渐地，计算机领域的会议论文也被认为是正式发表的作品，其重要程度至少不亚于期刊论文。

计算机期刊一般由某个出版社定期出版，可能是月刊或者季刊。期刊论文的投稿是没有截止日期的，随时都可以投。期刊的编辑通常会分配至少三名相关领域的专家进行评审。评审意见常常被分为四档：Accept without revision, Major revision, Minor revision, 和Reject。期刊论文在被录用之前，一般会经过数轮评审和修改。在每一轮评审中，评审专家会根据上一轮的评审意见和作者的修改再次给出修改意见。由期刊的编辑综合多位评审人员的意见来决定是否需要进行下一轮修改，或者拒稿。根据每轮论文需要的修改量，修改可分为Major Revision和Minor Revision两种。Major Revision一般涉及到重大修改，例如：修改论文框架，方法，或者实验内容。Minor Revision则一般指一些少量的文字或者描述上的修改。修改轮数完全由编辑决定（编辑会根据审稿意见决定），作者需要将论文修改到令编辑满意为止。

期刊每轮审稿的周期约为3-6个月，如果需要进行三轮评审的话，在被录用之前，一篇论文的审稿时间大约是1-1.5年。

有些期刊还会出版一些专刊（Special Issue）。这些专刊的征稿、评审、修改、和答复的过程与期刊一样，专刊专注于某一子领域，只接收该领域内的论文。



### 3.5 会议的CFP(Call For Paper)

一般国际会议在开放投稿之前会发布一份CFP (Call For Paper)。该文件会详细描述此次会议的目的、适用领域、会议的时间和地点、接收文章发行的刊物、接收文章的种类、接收文章研究的子领域(track)，以及投稿注意事项。通常会议文章种类包括，研究型文章(Research Paper or Regular Paper)，短文章(Short Paper)，和Demo文章(Demo Paper)。在投稿注意事项中，有以下几点需要研究人员仔细阅读。

1. 文章投稿系统。常用的有EasyChair，CMT等。
2. 文章投稿截至日期。有时会有两个截至日期。一个是摘要的截止日期，作者需在此日期之前注册文章，并上传完整的摘要内容。此时，并不需要上传完整的文章。第二个截至日期为投稿的截至日期，即在此日期前，作者需上传所有的内容（包括完整的文章和一些辅助材料，如果有的话）。
3. 文章的格式及长度要求。一般会议使用IEEE或者ACM的会议格式。但有些会议也使用自己维护和使用的文章格式。文章长度不能超过长度要求。
4. 是否给出作者列表。文章的审稿过程有open review, single-blind review和double-blind review三种方式。open review是指在审稿过程中，审稿人知道作者的身份，作者也知道审稿人的身份。single-blind review是指在审稿过程中，审稿人知道作者的身份，但是作者不知道审稿人的身份。这是最为常见的一种审稿方式。double-blind review是指在审稿过程中，审稿人不知道作者的身份，作者也不知道审稿人的身份。如果是此类审稿方式，作者不应在文章中给出作者列表。

有些会议还会提出一些更具体的要求，研究人员在投稿时需要详细阅读投稿要求，以免出错。

如何查找会议的CFP呢？最直接的方法是查看会议官方网站。一般会议每年举办一次。如果最新的CFP还没有发布的话，至少能够看到去年的CFP的内容。每年CFP的变化并不大。另外，WikiCFP和Call4paper收录了不少会议的CFP。如果暂时查找不到会议的官方网站，也可以在这两个网站试一试。

### 3.6 计算机科研领域分类

ACM (Association for Computing Machinery) 根据文章和其研究内容将计算机研究领域划分为13个子领域，它们包括

1. Topic A 通用文献(General and reference)，其主要涉及计算机领域内的通用话题，介绍性话题，调查报告和标准文档等内容。



2. Topic B 硬件(Hardware), 即硬件设计。其内容包括: 印刷电路板设计, 大规模集成电路设计和电子元器件设计等方面的内容。
3. Topic C 计算机系统结构(Computer Systems Organization)。其主要内容包括体系结构, 嵌入式系统, 实时系统等方面内容。
4. Topic D 网络(Networks)。其主要包含网络体系结构, 网络协议设计, 网络部件设计, 和网络算法设计等内容。
5. Topic E 软件工程(Software and its engineering)。该子领域包含软件组织结构, 编程技术, 软件管理等方面内容。
6. Topic F 计算理论(Theory of Computation)。该子领域主要研究计算模型, 形式语言与自动机理论, 语义与推理等。
7. Topic G 计算数学(Mathematics of Computing)。该子领域包括离散数学, 概率统计, 数值分析等内容。
8. Topic H 信息系统(Information Systems)。该子领域包括数据管理系统, 信息存储系统, 信息提取和万维网等。
9. Topic I 计算机安全(Security and Privacy)。该子领域包括加密, 安全理论, 入侵检测, 系统安全, 网络安全, 数据库安全等。
10. Topic J 人机交互(Human-centered Computing)。该子领域包括人机交互接口, 人机协作计算, 移动计算, 可视化等。
11. Topic K 计算方法学(Computing Methodologies)。计算方法学包括符号与代数操作, 并行计算方法学, 人工智能, 机器学习, 计算机图形学, 分布式计算方法学等。
12. Topic L 计算机应用(Applied Computing)。该子领域主要包含一些与计算机相关的交叉学科内容。例如: 电子商务, 企业计算, 科学计算, 生命医学计算, 计算机取证, 计算机教育等内容。
13. Topic M 社会与计算机行业话题(Social and Professional Topics)。该子领域包含计算工业与历史, 计算教育, 知识产权等。

更详细的信息可在 The 2012 ACM Computing Classification System 获取: [https://dl.acm.org/ccs/ccs\\_flat.cfm](https://dl.acm.org/ccs/ccs_flat.cfm)。

### 3.7 什么是SCI/EI索引期刊

SCI全称为Science Citation Index, 最初是由美国科学信息研究所(Institute for Scientific Information, ISI) 创办的文献检索文库。它收纳了自1900年以来, 150余个专业领域的8500多个世界著名的学术期刊。因为期刊筛选标准非常严格, 被SCI收纳的期刊常常被认为是具有世界领先水平的科学技术类期刊。





EI全称为Engineering Index。EI是目前世界上最大的工程类文献索引库。起初，因为查找文献非常困难，从事土木工程研究的约翰逊博士开始整理相关文献，并为这些文献撰写摘要，注明其文献的贡献。随后，在同行的帮助下，EI文献库慢慢发展壮大。在采用了计算机技术后，COMPENDEX（COMPuterized ENgineering inDEX）随之建立起来。目前，Compendex收录了两千多万条记录，每年新增长约八十万条记录，并由EngineeringVillage提供在线查询功能。

### 3.8 什么是影响因子(Impact Factor)

影响因子(Impact Factor or IF)是评价期刊影响力的一个重要指标。它表示着在过去两年内，期刊论文的平均被引用次数。然而，影响因子指标也是有其局限性的，因为一般研究热门问题的论文往往会比普通论文的引用次数高；研究应用问题的论文往往会比研究理论论文的引用次数高。在计算机领域，研究者们谈论的更多是会议或者期刊的档次。例如，top conference/journal 一般是指在第一档次的会议或者期刊。

### 3.9 论文提交管理系统EasyChair、Microsoft CMT和EDAS

论文提交管理系统是一套用于管理论文提交和评审流程的软件系统。在该系统中，作者可以提交论文，查看论文评审结果以及评审意见，提交论文的最终版本。审稿人则可以预览/下载待评审的稿件，提交评审意见。会议主席或者编辑可以邀请审稿人，分配稿件，以及发布最后的评审结果。

常用的会议论文提交管理系统为EasyChair、微软的CMT和EDAS。

EasyChair是一个基于web的会议管理系统。目前，已有四万多个会议采用了EasyChair提交和管理稿件。

另一个著名的学术会议管理系统是微软的CMT。到2016年为止，已有一千多个会议使用了CMT管理稿件。

EDAS是EDitor's ASsistant的缩写。EDAS早期用于管理IEEE/ACM Transactions on Networking。目前，EDAS仍然用于管理一些会议和杂志的论文提交与评审。

### 3.10 Microsoft Word与L<sup>A</sup>T<sub>E</sub>X之争

微软的Word和LaTeX（发音近似于雷泰赫）是常用的两种用于编写计算机学术论文的工具。

Word的优势在于所见即所得，使用方便，简单。但在格式调整方面，操作较为繁琐，且不易统一所有文章的格式。



LaTeX是基于TeX开发的一种扩展包。TeX实际上是一种编程语言。它通过定义一些指令，来控制文字/图片在页面中出现的位置，以及排版方式。但是因为TeX的复杂性，难以被人们直接使用，所以，LaTeX在TeX的基础上，扩展出了一些新的指令，降低了文档编辑的难度。虽然，使用LaTeX撰写论文需要一定时间的学习与练习，但是，这些都是值得的。因为，当一篇文章在IEEE和ACM的会议或者期刊之间转换格式时，作者只需切换论文模板文件，LaTeX会将文章自动调整为新的格式，为作者节省下大量的时间。

在计算机学术界，学者们更喜欢使用LaTeX撰写论文。与Word相比，LaTeX具有以下优势：

1. LaTeX能帮助生成复杂的数学公式。当作者熟练掌握LaTeX之后，编辑数学公式比Word更快，更高效。
2. LaTeX可以自动为章节，表格，图片，公式，脚注等标注序号。如果在文章插入或者调整章节，表格，图片，公式时，LaTeX会自动重新计算序号，无需人工调整。例如，在图二之前插入一张新图后，原图二自动变为图三，原图三自动变为图四，以此类推。
3. LaTeX可以自动更新章节，表格，图片、公式在正文中的引用序号，无需人工修改。例如，当原图二变为图三后，在正文中对该图的引用会自动更新。例如，“如图二所示”将被自动调整为“如图三所示”。
4. LaTeX可以自动生成参考文献列表。该列表中只包含文章中引用过的文献，而且可按照模板指定的格式自动生成参考文献。
5. LaTeX自动更新正文中参考文献序号。
6. LaTeX将文章的内容与格式分别定义在不同的文件中。作者编辑和提交的只是文章的内容，会议或者期刊的编辑会使用一个模板文件，将文章格式化为出版发行的格式。这样可以确保出版的文章均保持一致的风格与排版。

### 3.11 攻读硕士和博士期间应该学什么？

在攻读博士学位阶段，一名合格的博士生应着重学习和掌握两类知识和技能。

1. 专业知识和技能。从事科研工作，需要在某一领域，某一问题上投入大量的时间和精力来翻阅众多参考文献。阅读这些参考文献的过程实际上就是学习该领域的专业知识和技能的过程。
2. 认识问题的方法。在进行某项研究过程中，研究人员实际上是在反复进行着“发现问题-解决问题-再发现新问题-再解决新问题”。如何发现问题其实包含着几个子问题。例如：如何发现一个新问题；如何评估该问题的研究价值；和如何评价在该问题上已取得的成绩等。如何解决问题也包含着至少两个子问题：如何更好的解决该问题以推进科学技术的进步，和如何合理的验证新方法的有效性。经过多年反复的训练，研究人员会逐渐的建立起对这些问题的认识和个人见解，最终形成一种个人处理和解决问题的方法。这个方法是一种通用的方法，不仅可应用于计算机领域的科学研究之中，还可广泛的应用于其他领域的科研和其他日常工作学习之中。





以上两类知识和技能需要经过长时间的学习与锻炼，所以，博士研究生的学习过程较为艰苦漫长。然而，硕士研究生的学习时间较短，所以，其学习的着重点也略有不同。大致的讲，硕士研究生的主要培养的是锻炼解决问题的能力，即，给定一个问题，一名合格的硕士研究生需要能够自行查找文献并找到合理的解决方法，并且能够通过一定规模的实验来验证其方法的合理性和有效性。

### 3.12 研究人员应具备那些计算机技能？

笔者一直比较推崇“一个系统，一门语言”的目标，即至少熟练掌握一个操作系统/运行环境和一门通用型的编程语言。一个操作系统/运行环境指的是实验平台。由于绝大多数计算机领域的研究是需要利用实验结果来证明新方法的有效性的。所以，能够熟练使用实验环境是一项必备的技能。目前，大多数科研实验是在Linux系统下进行的。因此，学习Linux系统是一个不错的选择。

一门通用型的编程语言是指C/C++，Java，Python等通用编程语言中的一种。哪种语言更合适取决于所从事科研的领域，以及实验平台。笔者所从事的领域常常使用C/C++和Java。有时还会使用Python开发一些数据生成工具。在编程语言的选择过程中，笔者会考虑如下因素：

1. 运行环境/框架。如果所研究的内容十分依赖于某一个运行环境或者框架，那么，用于实现这一运行环境或框架的语言是最合适的编程语言。例如：笔者使用了C++语言实现了两个实验版本的文件系统。因为操作系统给出的文件系统接口是C语言的接口，而且C语言程序和C++语言程序能够被编译成一个模块，加载到操作系统中。在另一项目研究中，笔者使用了Java语言来修改Hadoop内部代码，以整合融入新算法。因为Hadoop框架使用的是Java，所以，笔者直接选择了Java语言。
2. 产出效率。因语言特性的不同，用其开发项目/实验程序的难度也不尽相同。如果语言的选择不太影响实验结果的话，应尽量使用自身熟悉的，开发周期快的语言。例如，笔者使用了Python开发实验数据生成器。因为实验数据生成的过程并不影响实验结果，而且Python具有开发速度快的优势，所以，笔者一般会优先考虑使用Python完成实验数据生成或者数据预处理，索引生成等工作。
3. 项目合作。如果一个项目是由多名研究人员共同完成的话，那么，选择一门大家都熟悉的编程语言比其他因素更重要。
4. 增量开发。如果从事的是基于某系统的增量开发，那么，保持使用同样的编程语言会使该系统具备较好的一致性。

### 3.13 如何解决Font Not Embedded问题？

在提交期刊文章的最终版本时，作者常常被要求提交一个pdf格式的论文文件，并需要通过期刊系统的检查才能最终提交成功。期刊系统的一个常规检查项是检查该pdf格式的论文文件是否已





#### 源代码示例

```
pdf2ps paper.pdf paper.ps
ps2pdf14 -dPDFSettings=/prepress -dEmbedAllFonts=true paper.ps paper-
All.pdf
```

如果Linux没有pdffonts命令，则请安装poppler-utils应用。

### 3.14 Regular Paper, Short Paper, Demo Paper 和Poster的区别

Regular Paper, Short Paper, Demo Paper和Poster是常见的四种会议论文类别。期刊并没有Short Paper, Demo Paper, 或者Poster。

总的来说，并没有一种规则来界定这四种论文类别。但是，大致上，Regular Paper或者Research Paper属于一种较完整的研究型论文。长度一般为8–12页。有时，较短一点的Regular Paper，可能只有6页。一篇Regular Paper需要包含所述科研题目相关的一切内容，例如：研究背景，新方法的特点以及效果等。

Short Paper属于一种较为简短的论文。因为篇幅原因，作者无法将所有的内容都涵盖在论文中，所以，在写作时，作者需要有所取舍，将最重要，最精华的内容放入短论文中。方法部分可以直接陈述结果，简化推导过程。实验部分可以直接表达最重要的结果。如果实在是篇幅太短，可将部分内容放在Technical Report中，在短论文中引用该Technical Report。

Technical Report并不是一种正式发表的论文。Technical Report的格式和内容完全由作者决定。Technical Report可以仅仅包含一个问题的定义，也可以仅仅包含一个方法的介绍，或者仅仅包含在论文中放不下的内容。Technical Report也没有正式发表的网络地址。作者可以将其放在学校的网站上，或者任意一个读者和审稿人能访问的地址上。

Demo Paper属于一种描述和表达一个系统的设计、实现、以及使用方式等内容。Demo Paper更偏向于应用，展示应用的效果。有时，一项科研成果可以转化为两篇论文。在Regular Paper中着重阐述科研成果的理论性和创新性，而在Demo Paper中侧重于描述其应用性。

Poster也属于一种描述和表达一项初步的科研成果或者一个系统的初步设计与实现。但是，区别在于Poster更加浓缩了这项成果或者系统的精华。Poster还可用于汇报项目的进展情况。所以，Poster所展示的内容是初步的，还处于预研阶段的。

### 3.15 论文写作对英语写作能力要求有多高？

英文论文写作或者英文科技文写作对于英语写作能力的要求是比较高的。笔者认为，这个要求在英语六级考试或者GRE写作考试的要求之上。但是，一篇论文毕竟不是一个语言上的艺术作



品，论文的写作在语言的用词造句方面并没有很高的要求，作者只需将论文的逻辑表述清楚即可。

对于中国人而言，在一篇上万字的论文中避免出现语言问题是很难的，而且费时费力。所以，当评审人在阅读文章时，如果只遇到少量的语法错误，但是这些错误并不影响阅读的话，那么，这篇文章就基本达到了论文的要求。

作为审稿人，在这里，笔者也谈谈审稿时读到语法错误的感受。一般来讲，在评审过程中，评审人员应保持客观、谨慎的态度，将重心放在文章的内容上，而不是语言问题上。然而，评审人和普通人一样，在审稿时，难免会有情绪上的波动。对于英语为母语的评审人而言，任何的语言错误都是非常“扎”眼的。语言错误很难逃过评审人的眼睛，而且一个小小的错误会影响到评审人的情绪。所以，如果一页仅仅出现一两个或少数语言错误的话，对于专业的评审人而言是影响不大的。但是，如果每一两行就会出现一个错误，那么，审稿人很可能会拒绝审理，甚至是直接拒稿。所以，在投稿前，需要仔细的检查论文中的语言问题。

那么，什么问题会让评审人无法接受呢？其一：句子不完整。如果出现不完整的句子，审稿人会认为这个稿件还没有达到发表的状态。其二：句子冗长，逻辑混乱，审稿人无法理解句子的含义。其三，由于语言或者符号错误，造成二义性。审稿人无法判断文章的正确性。其四：文章出现乱码。审稿人可以基本判断，在文章提交之前，作者并没有通读文章。

所以，笔者在此强烈建议，在论文提交之前，请将论文交由有经验的导师或者语言能力较强的同学、朋友通读一遍。在平时写作训练时，多留意观察其他优秀论文是如何用词和组织逻辑关系的。

### 3.16 如何写一封完美的Cover Letter?

在期刊论文初始提交时或者提交修改稿件时，作者往往需要准备一封Cover Letter，与论文一起提交到期刊系统中。那么，什么是Cover Letter呢？如何准备这封Cover Letter呢？

Cover Letter是一封简短的，描述此次论文提交内容的信。这封信用于向期刊的编辑和审稿人简要的介绍此次提交论文的目的，论文的内容等重要信息。站在期刊编辑的角度，在阅读完Cover Letter之后，期刊编辑需要做以下判断：（1）该论文是否符合期刊的要求，研究内容是否符合期刊所涉及的领域；（2）论文的贡献是否达到期刊的要求；（3）是否能找到三位审稿人审阅稿件。

所以，在Cover Letter中需要明确描述以下几点：（1）投稿目的，并列出论文题目和所投期刊的名称；（2）明确表明该论文是全新稿件，或者是由某会议接收论文的扩展成果。如果是某会议的扩展成果，需要明确指明会议的名称和时间，以便编辑查看论文的会议版本。最好能在提交时，将已发表的会议论文作为补充材料一起提交；（3）该论文的新贡献。如果该论文是某会议论文的延申，则Cover Letter需要明确指明该论文的新的贡献，以便编辑判断目前的论文能否独立成文。一般惯例是，期刊论文需要在会议论文的基础上新增至少30%以上，才能独立成为一篇论文。但是，如何衡量30%的新增量是一个非常主观的判断，而且，目前的趋势是30%的增量已不足以成为一篇新的论文，需要有更多新增的成果或者结论。



当此次提交的内容是回答审稿意见时，即在收到Major Revision或者Minor Revision的审稿意见后，作者对论文进行了修改，并将修改的结果答复给编辑和审稿人时，Cover Letter需要明确指出：（1）提交目的是为了回答此前Major Revision或者Minor Revision的修改建议；（2）简洁的总结此次修改的内容；（3）如果有某些地方未能修改或者无法修改，也需要提供合理的解释，以及相关证据的支持。

为了帮助读者更好的准备Cover Letter，小水滴在其网站相应的位置提供了Cover Letter的例子，供读者学习使用。

免责声明：小水滴提供的模板文件为笔者常年使用的Cover Letter的简化版本，仅供学习使用。在实际运用中，读者需按自身情况准备个性化的Cover Letter。

### 3.17 如果从其他领域“借鉴”一个方法，相关研究(Related Work)章节应该如何组织呢？

在科研过程中，“借鉴”其他领域的方法是一种常见的创新方式。那么，如果在具体的研究中，“借用”了其他领域的方法，相关研究(Related Work)章节应该如何组织呢？

在给出答案之前，我们需要回顾一下相关研究章节的目的。相关研究章节最主要的目的之一是展示论文的新颖性和独创性。为了合理而充分的论述论文的独创性，研究人员一般需要阐述两个关键点：研究现状和新方法的特点。事实上，研究现状的描述是为了说明在过去的研究中，该方法从未被使用过。而阐述新方法的特点其实就是着重论述新方法的独特之处，以及新方法为什么能解决某些问题，而已有的方法却不能解决这些问题。

因此，回到最初的问题上来。当从其他领域“借鉴”了一个方法时，研究人员首先需要详细描述该方法在其他领域是如何被使用的，这些相关的研究结果展示了哪些该方法的优势和缺点。然后，第二步是描述该论文所研究领域的现状，以及亟待解决的问题。如果该方法的优势正好能被使用来解决所研究领域内的问题，那么，将该方法引入这个领域是一件顺理成章的事情了。所以，在写相关研究章节之前，研究人员需要事先确定所“借用”的方法的优势，以及为什么需要将其引入所研究的领域。如果这两点能够非常明确的列出来，那么，上述的三层逻辑也就很容易衔接起来了。

### 3.18 如何在阅读论文的过程中发现新的研究问题？

如何发现新的研究问题是每位科研人员都会遇到的问题。其实，这个问题包含了两层意思。其一，需要发现一个从未被发表过的问题；其二，这个问题是一个值得投入研究的问题。在这两点之间，相对而言，发现一个值得研究的问题更加困难。因为，经过一段时间的文献阅读和文献搜索，研究人员应能初步判断一个问题是否已被研究和发表。然而，对于第二点而言，“是否值得研究”是一个非常主观的，因人而异的问题。





判断一个问题是否值得研究大致依赖于两个关键点，即该问题的创新点以及该创新点对科研或者人类社会做出的贡献的大小。此处，我们着重阐述识别和发现创新点。对于贡献大小的评估，大致可分为四个方面（影响力、新颖性、理论深度、和有效性）。关于这方面的介绍可参考小水滴论文写作的前言部分。

创新点的识别是一名科研工作者非常重要的一项能力。绝大多数论文的新颖性来自于两个方面：问题的新颖性和方法的新颖性。问题的新颖性是指论文所研究的问题是一个新的问题。这里的“新”可以是新的问题定义，新的运行环境，新的输入输出，新的处理流程，或者新的假设等。我们利用以下几个例子来一一解释这些“新”的内容。

1. 新的问题定义：一项科研工作需要有一个明确的科研目标或者问题的定义。例如，在一项数据中心的节能研究项目中，研究人员需要明确的定义如何计算或者测量能量的消耗。如果采用实际测量的方法获得能量消耗的数值，那么，研究人员需要明确定义哪些部件需要测量。如果测量方式不同或者测量对象不同，我们可以认为基于不同测量方式和测量对象的研究问题是不同的问题。它们各自具有独特性。
2. 新的运行环境：这一项比较容易理解。单机算法和并行算法的研究是不同的。并行算法和分布式算法的研究也是不同的。在分布式算法中，如果网络结构不同，那么，所研究的问题也是不同的。它们各自具有独特性。
3. 新的输入输出：当某一项研究的输入输出数据不同时，这项研究的问题可能会变成完全不同的另外一个问题。例如：在分布式计算模型中，各个计算节点（资源）常常被视为一种输入数据，那么，基于同构系统的研究和基于异构系统的研究常常是完全不同的问题，它们各自具有独特性。
4. 新的处理流程：如果研究对象是某一处理流程，那么，如果该处理流程变化了，其研究方法和结论也可能随之变化。即使处理流程不变，而研究的着眼点发生了变化，那么，其研究方法和结论也可能随之变化。例如，着眼点从局部优化变成了全局优化，或者着眼点从CPU-Intensive的步骤转移到I/O-intensive的步骤。它们各自具有独特性。
5. 新的假设：实际上，问题定义、运行环境、输入输出、和处理流程都是一种问题的假设。但是，在这一点上，我们将解释一些更加明显且具体的假设问题。例如：目前有科研人员研究如何使用无人机帮助发送包裹的研究。有一些科研人员主张一辆邮递车可搭载一台无人机，并且假设在停车时，该无人机可由司机操作，发送包裹。另一批科研人员则主张使用智能无人机，他们假设当司机设置好无人机后，无人机可自行投递包裹，并在下一个目的地等待司机。在这个过程中，司机不必等待无人机返航。司机可驾驶邮递车，前往下一个目的地。在该类问题上，还存在很多其他不同的假设，例如，一辆邮递车可搭载多台无人机等。它们各自具有独特性。

如果所研究的问题完全相同，那么，科研人员应该能够很容易的找到论文中新方法的独特性。此时，多留意思考该新方法提出的动机、解决问题的思路、以及背后的原理，并将其记录下来。这些都有助于探索发现新的研究问题。



总之，识别和发现新的研究问题是一种科研能力，培养这种能力需要经过长时间的积累，平时应注意多思考，勤练习。

### 3.19 如何应对会议/期刊论文被拒？

对于研究人员而言，论文被会议/期刊接收(Accept)或者拒稿(Reject)是一件很正常的事情。若论文被接收，固然好；但是，如果论文被拒稿，也无需气馁，因为，论文被拒稿的原因有很多。本文着重介绍笔者的经验，以及应对方法。

首先，研究人员需要对投稿论文的质量，以及所投会议/期刊期待可接收的论文质量有所了解。如果所投稿件不能达到会议/期刊的期望的话，或者稿件的内容不符合会议/期刊的研究领域和内容的话，投稿命中率是不会高的。

其次，研究人员需要仔细分析审稿意见，改进稿件的质量，以便于提高下一次投稿的命中率。一般来说，审稿人应是该领域或者相关领域的、经验丰富的专家，他们给出的审稿意见大部分是具有建设性的、值得认真思考的。笔者根据自身的经历与经验，将审稿意见分为以下几类。

1. 第一类：针对论文方法上的，结构上的，实验结果上提出的疑问。此类问题是需要仔细思考的。如果论文在方法上、算法上存在错误，那么，此论文基本上是不可能被高质量会议/期刊录用的。如果是论文的结构上有问题，那么，在结构上的改进可以大大提升论文的质量。因此，研究人员应在结构问题上多花时间。对于实验结果的疑问可能出于多种原因，最严重的一种是实验结果与使用方法的理论结果不一致。这有可能是新方法存在问题，也有可能是实验设计或者实施上存在问题，需要仔细分析。
2. 第二类：针对论文动机(Motivation)提出的疑问。如果审稿人认为论文的动机没有表达清晰的话，这种问题相对容易解决。论文动机的写作方法可参考论文写作第四章概述章节的内容。如果论文的动机已表达清晰，但是，审稿人认为该动机不足以引出一项研究项目的话，解决这种问题会复杂一些。因为，论文影响力的评估是一个非常主观的，因人而异的问题。解决此类问题的一种常见思路是引用一些论文或者商业/科研项目，用以支持阐述开展本项研究的必要性。
3. 第三类：针对论文表述的新方法有效性提出的疑问。如果审稿人质疑的是新方法的理论分析的有效性的话，论文改进的空间可能不会很大，因为这可能涉及到需要重新设计或者改进新方法，从而大幅度的修改论文。如果审稿人质疑的是实验结果表现出的有效性的话，研究人员需要思考一下实验的设计，如何更好的表现出新方法的优势。
4. 第四类：针对论文中的细节部分提出疑问。此类问题比较容易解决。如果仅以文字描述无法表达清楚的话，添加图、表、和公式是常见的增加表达力的方法。
5. 第五类：遇到无法解决的问题的时候，研究人员需要多思考无法解决的真实原因。从这些原因当中，研究人员可能可以发现论文中潜在的缺陷。问题无法解决的原因包括：新方法存在缺陷、问题不在本论文讨论范围之内、实验数据集无法取得或者收集、算法/系统无法实现、实验环境限制等。





- 第六类：审稿人的审稿意见错误。有时审稿人是会犯错误的，因为他们审稿的时间有限。然而，这种情况并不常见。有时，研究人员可能认为审稿人没有完全理解论文的内容。但是，这也可能是因为审稿人是站在更高的一个层次在审视这个问题而导致的。所以，当研究人员认为审稿人的意见出错时，应多与导师讨论。

总之，论文被拒是一件常见的事情。在论文被拒后，研究人员应多思考如何改进论文。

### 3.20 科研人员需要了解哪些会议/期刊信息？

作为一名科研人员，跟踪和掌握会议和期刊的信息是一项基本的任务和技能。那么，会议和期刊的哪些信息是科研人员必知必会的呢？

- 会议/期刊的科目分类。因从事的领域不同，科研人员需要了解的会议和期刊也不相同。例如：计算机系统这个领域，科研人员研究的是系统的架构，包括操作系统的架构、并行/分布式计算系统的架构，网络系统的架构等。（有时，计算机网络也可看作是一个独立的子领域。）那么，在这个领域内，著名的会议有Usenix OSDI, ACM SIGCOMM, ACM SIGOPS等。著名的期刊有 ACM Transactions on Computer Systems (ACM TOCS), IEEE Transactions on Parallel and Distributed Systems (IEEE TPDS)等。科研人员需对科目分类熟记于心。
- 会议/期刊的质量。在计算机领域，科研人员更喜欢使用会议/期刊的级别来界定其质量。（计算机领域很少使用影响因子。）计算机会议/期刊级别的排定也有着不同的标准。但是，总的来说，顶级会议和期刊的座次大致相同。有关计算机会议/期刊的科目分类和排名可查阅这里。
- 会议/期刊的名字以及英文缩写。科研人员需要非常了解领域内的顶级会议/期刊的名字以及英文缩写，因为在平时的对话和讨论中，科研人员更多的是使用英文缩写，这样更方便、易于交流。科研人员还需要大致了解其他领域内的著名的顶级会议和期刊名字。一般来说，Usenix的会议，ACM的会议和期刊(ACM Transactions)，IEEE的会议和期刊(IEEE Transactions)，Springer和Elsevier的期刊的质量是很高的。
- 会议的时间。高水平的会议一般在较为固定的时间（月份）召开。例如：Usenix OSDI一般在10-11月召开，隔年举行。ACM SIGCOMM每年在8月份举行。
- 会议/期刊接收文章的领域。这是指会议/期刊接收什么领域的文章。在会议Call For Paper或者期刊的主页上会给出它们感兴趣领域的列表。
- 会议/期刊接收文章的质量。顶级会议/期刊对文章的要求是很高的。通过阅读它们的文章可以对其质量的要求有大致地了解。如果需要完全了解的话，建议向会议/期刊直接投稿，经历过一两次投稿和评审之后，才能真正了解顶级会议/期刊竞争的激烈程度，毕竟，被接收的文章只是高水平文章中的极少一部分。
- 会议/期刊的论文格式。Usenix, ACM, IEEE, Springer和Elsevier都维护了一套论文格式。所以，如果研究人员知道了会议的组织机构的话，自然也就知道其论文的格式了。



## 4 L<sup>A</sup>T<sub>E</sub>X常用的例子

本文主要介绍在计算机论文写作中需要用到的L<sup>A</sup>T<sub>E</sub>X的知识和一些常见用例。因为T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X是一个非常复杂的编辑系统，本文无法涵盖所有的知识内容。若对T<sub>E</sub>X感兴趣，可参考T<sub>E</sub>X作者（图灵奖获得者Donald Knuth）撰写的《The T<sub>E</sub>Xbook》。LaTeX的相关文档可以在其官方网站([www.latex-project.org](http://www.latex-project.org))上找到。《The Not So Short Introduction to LaTeX》也是一本很好的入门书籍。

### 4.1 环境搭建

LaTeX工作环境的搭建分为两个步骤，即选择LaTeX编译环境和编辑环境。

在Windows平台上，MiKTeX和TeX Live是最为流行的两个编译环境。它们都包含完整的T<sub>E</sub>X系统的工具，例如：pdfTeX，XeTeX，LuaTeX和BibTeX等。但是，它们也有一些区别。MiKTeX默认仅安装最基本的包，当所编辑的文档需要新的包时，MiKTeX会自动地下载并安装所需要的包。TeX Live也提供基本安装包和全部安装包。笔者认为可根据个人爱好选择其中一种。在Linux平台下，TeX Live提供了Debian/Ubuntu包，可直接使用系统包管理器安装（`apt-get install texlive`）。

LaTeX的编辑环境则有更多的选择。笔者经常使用Texmaker完成LaTeX文档的编辑工作。当TexMaker安装完成后，用户可能需要在选项->配置Texmaker中设置LaTeX编译环境的路径。

另一个比较流行的在线LaTeX编辑环境是Overleaf。它提供多用户之间文档共享，多用户同时联合编辑，和云存储文档等服务。

### 4.2 文章结构

如果使用LaTeX撰写文章，那么，文章一般是从一个空白模板开始的。LaTeX的模板（包括会议论文模板和期刊论文模板）一般会包含以下几个部分。

1. 模板文件以`\documentclass{options}{class}`开始。`\documentclass`定义了论文的格式布局，以及一些全局性的设置。例如，字体大小，单双栏格式，和纸张大小等。
2. 随后，模板文件导入包(packages)。一个包实际上是一个命令库，它提供了一些格式化的功能（例如，格式化算法中的伪码）。第一二部分被称为模板的前言。
3. 从第三部分开始，模板文件正式进入正文部分。文章的正文是包含在`\begin{document}`和`\end{document}`之间的内容。
4. 接下来的是`\title`和`\author`。不同的模板，`title`和`author`可能会稍有不同。模板往往会使用`\maketitle`，以用于显示标题。



5. 摘要需放置在`\begin{abstract}`和`\end{abstract}`之间。有些模板会要求提供关键字信息（在`\begin{IEEEkeywords}`和`\end{IEEEkeywords}`之间）。有些模板则没有这项内容。
6. 最后是论文的主体部分，其中，一般包括概要章节，相关研究章节，设计章节，实验章节，结论章节，致谢章节和文献列表。

以下是IEEE会议模板的组织结构。该例子主要用于展示LaTeX论文模板的结构，所以忽略了一些IEEE会议模板特有的命令。

#### 源代码示例

```
\documentclass[conference]{IEEEtran}
\usepackage{algorithmic}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}
\title{Paper Title}
\author{\IEEEauthorblockN{1\textsuperscript{st} Given Name Surname}
\IEEEauthorblockA{\textit{Department of Computer Science} \\\
\textit{XXX University} \\\
City, Country \\\
email address}
}
\maketitle
\begin{abstract}
This is an abstract.
\end{abstract}
\begin{IEEEkeywords}
Keyword Examples
\end{IEEEkeywords}
\section{Introduction}
This is an introduction section.
\section{Related Work}
This is a related work section.
\subsection{Design}
This is a design section.
\subsection{Experimental Validation}
This is an experimental validation section.
\subsection{Conclusion}
This is a conclusion section.
\section*{Acknowledgment}
This is an acknowledgment.
\bibliographystyle{plain}
```



```
\bibliography{reference}  
\end{document}
```

注：`\section{}`和`\section*{}`的区别在于，`\section{}`会将该章节名称添加入目录(table of contents)和导航栏(navigation bar)中；而`\section*{}`仅仅将章节名称添加入导航栏(navigation bar)。这个小小的区别对于论文的撰写没有影响。如果需要，会议或者期刊编辑会加上的。

### 4.3 文件结构

由于一篇论文篇幅较长，如果将所有内容都写在一个文件里，这个文件的文本内容会非常长。当需要修改时，查找定位待修改的内容会较为困难。以下是笔者较为推荐的方法。

为了减轻这个查找和定位的难度，论文主体部分的章节可以写在单独的文件中，通过`\include`将其包含在主体文章中。

如下是一个文件组织结构的例子。其主文件是`paper.tex`。如图 9所示，主文件使用`\include`命令将各个章节的内容从其各自的文件中导入。在使用`\include`时，被导入文件的后缀`.tex`可省略。`\label`命令是给各个章节一个标签，可以在文章的其他地方引用该章节序号。参考文献列表章节由`\bibliography`命令引入。`\bibliographystyle{abbrv}`指明了参考文献列表以缩略的形式出现，用以压缩参考文献列表的长度。该例中参考文献列表采用较小的字体。

图 10展示的是该篇论文的文件结构。从图中可以看出，除了主体文件`paper.tex`和各章节的文件以外，`IEEEtran.cls`是IEEE的模板文件。该文件定义了IEEE文章中各种格式化的参数。使用该文件，可将论文格式化为IEEE的标准格式。`paper.pdf`为最终生成的pdf版本的论文。`reference.bib`为参考文献列表文件，在主体文件中由`\bibliography`命令引入。值得注意的是，因为LaTeX只会导入论文主体中引用过的参考文献，所以，每位研究人员只需要维护一个参考文献全集文件（即可将感兴趣的文献都包含在这个文件中）。在LaTeX编译过程中，文章未使用的文献不会出现在文章中。`figs`文件夹里面存放的是论文使用的图片文件。`Initial Submission`，`Major Revision-1`和`Minor Revision-2`文件夹则存放着此论文修改的各个版本。

### 4.4 生成文章PDF版本

如果使用LaTeX撰写会议或者期刊论文，一般会要求提交一个PDF格式的论文文件。期刊则在最后会要求提交论文的源代码文件。这个PDF格式的文件是由一个复杂的编译过程生成的。这个过程一般被称为“LaTeX+BibTeX+LaTeX+LaTeX”，或者“LaTeX+BibTeX+LaTeX(x2)”。这是因为这个编译过程需要先运行LaTeX一次和BibTeX一次，然后再运行LaTeX两次。如果不按照上述方式运行，则编译出来的论文可能会包含错误的（章节、表、图、公式的）引用序号。最后运行一次PDFLaTeX，即可得到PDF版本的论文。

那么，为什么需要重复的执行LaTeX呢？这是因为这些步骤需要计算章节/图/表/参考文献的



```

% make the title area
\maketitle

\section{Introduction}\label{sec:intro}
\input{intro}

\section{Related Work}\label{sec:related}
\input{related}

\section{Preliminaries}\label{sec:prelim}
\input{prelim}

\section{Design}\label{sec:design}
\input{design}

\section{Experimental Validation}\label{sec:expr}
\input{expr}

\section{Conclusion}\label{sec:conc}
\input{conc}

%\section*{Acknowledgements}
%\input{ack}

\small
\bibliographystyle{abbrv}
\bibliography{reference}

\input{Short-bios.tex}
\end{document}

```

图 9: 主体文件的内容。

Name	Date modified	Type
figs	10/6/2018 11:07 PM	File folder
Initial Submission	10/6/2018 11:07 PM	File folder
Major Revision-1	10/6/2018 11:07 PM	File folder
Minor Revision-2	10/6/2018 11:07 PM	File folder
ack.tex	7/23/2012 10:38 A	(La)TeX document
conc.tex	8/9/2015 8:55 PM	(La)TeX document
design.tex	8/10/2015 9:47 PM	(La)TeX document
expr.tex	8/14/2015 9:27 PM	(La)TeX document
IEEEtran.cls	12/15/2009 8:45 A	(La)TeX Class File
intro.tex	6/14/2015 5:18 PM	(La)TeX document
paper.pdf	10/2/2014 3:34 PM	Adobe Acrobat D...
paper.tex	8/14/2015 9:28 PM	(La)TeX document
prelim.tex	8/10/2015 9:36 PM	(La)TeX document
reference.bib	2/19/2015 11:27 A	(La)TeX Bibliograp...
related.tex	8/10/2015 9:35 PM	(La)TeX document
Short-bios.tex	8/9/2015 5:34 PM	(La)TeX document

图 10: 文件的组织结构。

序号和引用位置，最后再将它们插入论文中。其各步骤的具体细节为：

1. 第一次运行LaTeX，它会将正文中引用文献的地方（`\cite{}`）写入.aux文件。
2. 第一次运行BibTeX，它会读取.aux文件，然后将使用到的文献引用写入.bbl文件。
3. 第二次运行LaTeX，它会将.bbl文件的内容插入到`\bibliography{}`的地方，并更新.aux中的label信息。
4. 第三次运行LaTeX，它会将.aux中已更新的label信息插入到原文中。这样原文中就能显示文献引用和序号了。

## 4.5 章节

使用LaTeX可以很容易的将论文内容划分为章节，子章节，和子子章节三个层次。这三个层次可分别由 `\section{}`，`\subsection{}`和`\subsubsection{}`标注。章节的编号一般保持默认的风格即可。有的模板使用阿拉伯数字，有的模板使用罗马数字。

以下是一个使用当前IEEE期刊格式的章节列表用例（如图 11所示），它使用的是阿拉伯数字。

在源代码中，`\label{sec::section-2}`指定了第二章的标签名称为sec::section-2。所以，在章节1.1中，可使用 `\ref{sec::section-2}`引用第二章的序号。注意，第一个字符“波浪线” (tilde)是标准键盘左上角上边的符号。在TeX/LaTeX中，波浪线是一种特殊的空格，在论文撰写中，常与章节，图，表和参考文献引用同时使用。



## 1 SECTION 1

### 1.1 SubSection 1.1

主要内容见第 2 章。

#### 1.1.1 Subsubsection 1.1.1

#### 1.1.2 Subsubsection 1.1.2

#### 1.1.3 Subsubsection 1.1.3

### 1.2 Subsection 1.2

#### 1.2.1 Subsubsection 1.2.1

## 2 SECTION 2

### 2.1 Subsection 2.1

#### 2.1.1 Subsubsection 2.1.1

图 11: 章节列表用例。

#### 源代码示例

```
\section{Section 1}\label{sec::section-1}
\subsection{SubSection 1.1}\label{sec::section-1.1}
    主要内容见第~\ref{sec::section-2}章。
\subsubsection{Subsubsection 1.1.1}\label{sec::section-1.1.1}
\subsubsection{Subsubsection 1.1.2}\label{sec::section-1.1.2}
\subsubsection{Subsubsection 1.1.3}\label{sec::section-1.1.3}
\subsection{Subsection 1.2}\label{sec::section-1.2}
\subsubsection{Subsubsection 1.2.1}\label{sec::section-1.2.1}
\section{Section 2}\label{sec::section-2}
\subsection{Subsection 2.1}\label{sec::section-2.1}
\subsubsection{Subsubsection 2.1.1}\label{sec::section-2.1.1}
```

注：这里有个小诀窍。在标签命名时，标签名是不能重复的，即章节，图，表，公式和参考文献的标签名不能重复。如果重复的话，LaTeX会用后出现的标签名称覆盖前面出现的标签名称，所以，文章中的引用序号会出现错误。如果在标签名中加入一个固定的前缀，以表示引用对象的类别，可以减小出现重复名称的概率。例如，章节的标签均以sec::为前缀，图的标签均以fig::为前缀，表格的标签均以tab::为前缀，公式的标签均以equ::为前缀，参考文献的标签均以ref::为前缀。

## 4.6 图

在计算机论文中，常常出现五种图片排版的方式。它们是，单栏单图格式，跨栏单图格式，两种双图并排单栏格式，和多图跨栏格式。图的标题出现在图的下方。在`\begin{figure}`指令之后，可以使用一个可选的参数指定该图放置在页顶（`\begin{figure}[t]`）或者页底（`\begin{figure}[b]`）。





默认值则指定图片按照论文内容的先后顺序放置。

4.6.1 单栏单图格式

最为常见的一种图片排版格式是单栏单图格式。无论是单栏格式的论文或者双栏格式的论文，单栏单图格式将一张图完全占满当前栏的空间。图 12展示了一个在一篇双栏格式的论文中，使用了单栏单图格式放置图 1。图 1占满了左栏的整个空间。

注：图的标题一般出现在图的下方。

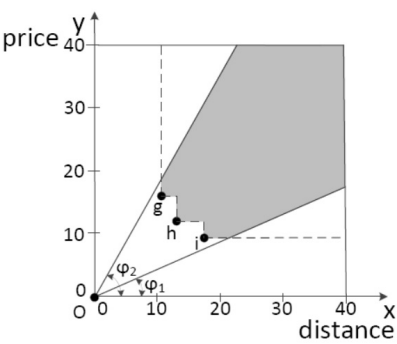


Fig. 1. An example of angle-based partitioning.

indexing services. Names should not be listed in columns nor group by affiliation.

G. Overview

This article introduces preliminaries of paper writing in computer science, and presents research experience and skills in manuscript preparation, revision, and submission. All examples in this article are selected from papers published in top conferences or journals. Although there are a variety of approaches to organize a research paper, this article and examples focus primarily on demonstrating methods usually adopted by the author of this article in paper writing. We hope that these materials will be helpful to researchers who are working hard to establish their research career.

There are many types of research papers. The one this article will present is called **Research Paper** or **Regular paper**.

图 12: 单栏单图的例子。

在论文中显示图片，首先需要导入graphicx包。然后在figure环境中（在\begin{figure}和\end{figure}之间），将图片居中对齐（\centering），设置图片文件位置和大小（\includegraphics），给出图片标题（\caption）和设置图片标签（\label）。

源代码示例

```
\documentclass[10pt,journal,letterpaper]{IEEEtran}
\usepackage{graphicx}
...
\begin{figure}[h!]
  \centering
  \includegraphics[width=0.5\textwidth]{figs/single_figure.jpg}
  \caption{单图的例子。}
  \label{fig::single_figure}
\end{figure}
```

4.6.2 跨栏单图格式

在双栏格式的论文中（例如IEEE或者ACM会议论文），如果图较大，无法放入单栏时，可考



虑同时使用两栏，以跨栏的方式放置该图。这种情况常常出现于展示系统框图或者流程图。在下图中，该论文是一篇双栏的IEEE的期刊论文。由于图中的图 1是一副系统流程图，描述了整个系统的输入输出和整个处理过程的两个阶段。所以，为了能够清晰的展示该图，作者使用了跨栏单图格式，给与足够的展示空间。

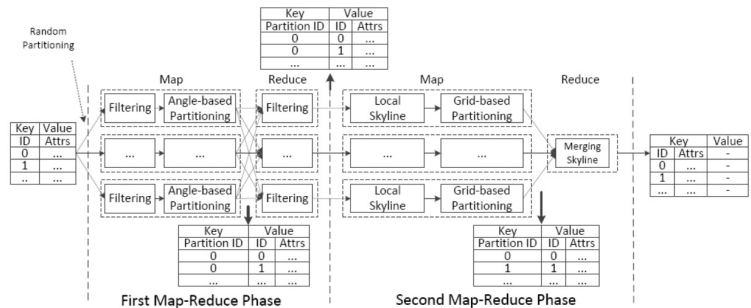


Fig. 1. An overview of the parallel skyline processing using MapReduce.

in manuscript preparation, revision, and submission. All examples in this article are selected from papers published in top conferences or journals. Although there are a variety of styles in manuscript preparation, revision, and submission. All ex-

This article introduces preliminaries of paper writing in computer science, and presents research experience and skills in manuscript preparation, revision, and submission. All ex-

图 13: 跨栏单图的例子。

如下是跨栏单图例子的源代码。这里用到了figure\*环境，而不是单栏单图格式使用的figure环境。其他的内容与单栏单图的例子类似。

源代码示例

```
\documentclass[10pt,journal,letterpaper]{IEEEtran}
\usepackage{graphicx}
...
\begin{figure*}[h!]
  \centering
  \includegraphics[width=0.75\textwidth]{figs/single_figure_col_span.jpg}
  \caption{跨栏单图的例子。}
  \label{figure:single_figure_col_span}
\end{figure*}
```

4.6.3 单栏双图并排格式

使用单栏双图并排格式有着两个目的。其一是将两个相近的图放在一起，可以做比较。其二是为了节省论文空间。有些论文是有长度限制的，如果论文内容超过了这个限制，将两个图并排展示是一个快速缩短论文长度的办法。当然，双图并排格式缩小了图片展示的空间。所以，有些图并不适合使用这种格式。

单栏双图格式有两种形式。在下图中，例文中的图 9使用了两张图（9.a和9.b）展示了在两种数据集上的实验结果。这两张图形成了强相关的并列关系。

iterations.

- 2) This article introduces preliminaries of paper writing in computer science, and presents research experience and skills in manuscript preparation, revision, and submission. All examples in this article are selected from papers published in top conferences or journals. Although there are a variety of approaches to organize a research paper, this article and examples focus primarily on demonstrating methods usually adopted by the author of this article in paper writing. We hope that these materials will be helpful to researchers who are working hard to

create an object in the closest partition are returned first. The order of objects in a particular partition is

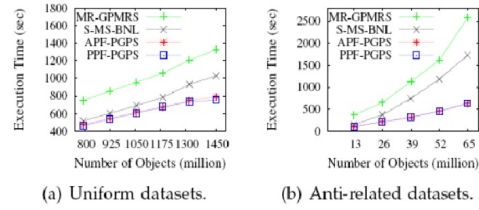


Fig. 9. Varying dataset cardinality.

图 14: 单栏双图并排格式 (例一)。

在图 9 的源代码中, 该例引入了 subfigure 包, 并使用了 subfigure 环境, 将图缩小。值得注意的是, 整个图组 (图 9), 两个子图 (图 9.a 和 9.b) 具有独立的标签。换句话说, 它们能在正文中被单独引用。比如, 在解释整个图组实验结果时, 可以使用标签 fig::card。而解释在 Uniform datasets 上的实验结果时, 可以使用标签 fig::card\_uni。

#### 源代码示例

```
\documentclass[10pt,journal,letterpaper]{IEEEtran}
\usepackage{graphicx}
\usepackage{subcaption}
...
\begin{figure}[h!]
  \centering
  \subfigure[Uniform datasets.]{
    \label{fig::card_uni}
    \includegraphics[width=1.53in]{figs/expr/Card_uni.eps}}
  \subfigure[Anti-related datasets.]{
    \label{fig::card_anti}
    \includegraphics[width=1.53in]{figs/expr/Card_anti.eps}}
}
\caption{Varying dataset cardinality.}
\label{fig::card}
\end{figure}
```

第二种单栏双图并排格式是将两张独立的图放在一起, 如下图所示, 图 14 和图 15 描述的是完全不同的内容, 可以单独的以单栏单图的格式展示。然而, 在该论文中, 作者为了节省文章的空间, 将两张图并排展示, 以将更多的论文空间留给其他更重要的部分。

从本例的源代码看, 该例在 figure 环境中使用了 minipage 环境, 将两张图并排展示。

#### 源代码示例



iterations.

- 2) This article introduces preliminaries of paper writing in computer science, and presents research experience and skills in manuscript preparation, revision, and submission. All examples in this article are selected from papers published in top conferences or journals. Although there are a variety of approaches to organize a research paper, this article and examples focus primarily on demonstrating methods usually adopted by the author of this article in paper writing. We hope that these materials will be helpful to researchers who are working hard to

organize their objects in the closest partition as possible first. The order of objects in a particular partition is

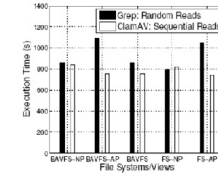


Fig. 14. Random and sequential read evaluation in BAVFS.

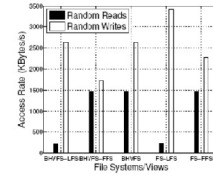


Fig. 15. Random read and write evaluation in BHVFS.

图 15: 单栏双图并排 (例二)。

```
\documentclass[10pt,journal,letterpaper]{IEEEtran}
\usepackage{graphicx}
...
\begin{figure}[h!]
  \begin{minipage}{0.46\linewidth}
    \centering
    \includegraphics[width=1.5in]{figs/partial_order.eps}
    \caption{Grid-based partition partial order in a two dimensional space.}
    \label{fig:partial}
  \end{minipage}
  \hspace{0.01\textwidth}
  \begin{minipage}{0.46\linewidth}
    \centering
    \includegraphics[width=1.5in]{figs/grid_partition.eps}
    \caption{Sentinel objects.}
    \label{fig:sentinel}
  \end{minipage}
\end{figure}
```

#### 4.6.4 跨栏多图格式

跨栏多图格式是将整个论文的宽度分配给多个图，以期取得最好的展示效果。跨栏多图格式常常用于展示实验结果。例如，一行展示三个相关的实验结果图。

图 16 中，三个图并排排成一行。其中，图 16a 和 16b 展示的是学校 (schools) 和超市 (supermarkets) 的两个例子。图 16c 则是图 16a 和 16b 叠加的效果图。这三个子图组成图 16。

在源代码中，0.28 是三个子图的宽度，占整个文章宽度的 28%。其他设置类似于其他图的格式设置。



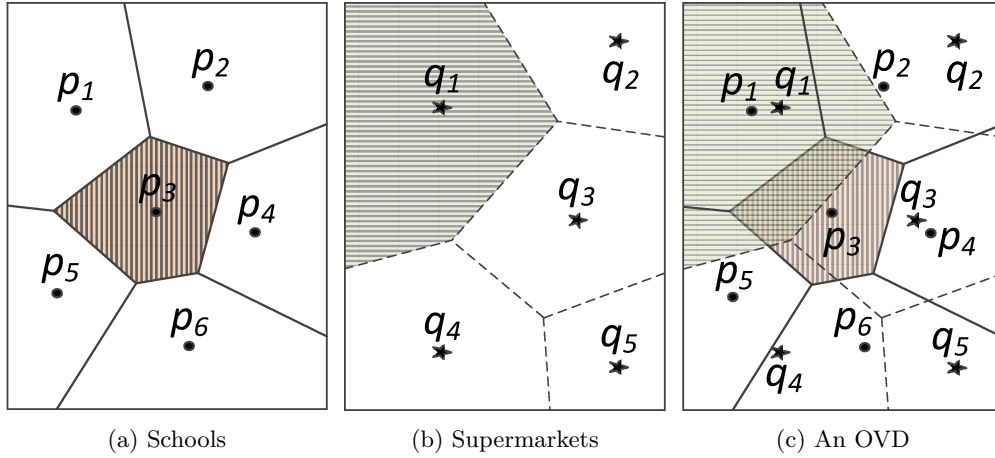


图 16: Ordinary Voronoi diagrams and OVDs in Euclidean space.

#### 源代码示例

```
\documentclass[10pt,journal,letterpaper]{IEEEtran}
\usepackage{graphicx}
\usepackage{subcaption}
...
\begin{figure*}[h!]
  \centering
  \begin{subfigure}[h!]{0.28\textwidth}
    \includegraphics[width=\textwidth]{figs/figures_col_span_1.eps}
    \caption{Schools}
    \label{fig::figures_col_span_1}
  \end{subfigure}
  ~
  \begin{subfigure}[h!]{0.28\textwidth}
    \includegraphics[width=\textwidth]{figs/figures_col_span_2.eps}
    \caption{Supermarkets}
    \label{fig::figures_col_span_2}
  \end{subfigure}
  ~
  \begin{subfigure}[h!]{0.28\textwidth}
    \includegraphics[width=\textwidth]{figs/figures_col_span_3.eps}
    \caption{An OVD}
    \label{fig::figures_col_span_3}
  \end{subfigure}
  \caption{Ordinary Voronoi diagrams and OVDs in Euclidean space.}
  \label{fig::figures_col_span}
\end{figure*}
```



## 4.7 表格

在论文中，常见的表格有两种形式，即单栏表格和跨栏表格。表格的标题出现在表的上方。表格常用于列举或者对比某一类事物。表格除了可以在`\begin{table}`的指令中设置页顶显示或者页底显示（与图的设置类似）以外，表格还可以设置单元格内的对齐方式。在`\begin{tabular}`指令中，每一个字母表示一栏，l表示该栏左对齐，c表示居中对齐，r表示右对齐。p{‘width’}，n{‘width’}和b{‘width’}则表示单元格在垂直方向向顶对齐，居中对齐，和向底对齐。一个竖线|则表示在两个相邻的单元格之间显示一条竖线，以分割单元格。

### 4.7.1 单栏横向表格

顾名思义，单栏表是展示在单栏中的表格。如果是在双栏格式的论文中，单栏表只会出现在左栏或者右栏中。单栏横向表格由于受到宽度的限制，栏数不能太多。例如：下图中的表1比较了方法A和方法B的运行时间（单位：秒）和输出结果（距离）的数值。

ples in this article are selected from papers published in top conferences or journals. Although there are a variety of approaches to organize a research paper, this article and examples focus primarily on demonstrating methods usually adopted by the author of this article in paper writing. We hope that these materials will be

表 1: Method Comparison.

	时间（秒）	距离
A	83.6	354
B	80.2	402

图 17: 单栏横向表格。

#### 源代码示例

```
\begin{table}[h!]  
  \centering  
  \caption{Method Comparison.}  
  \label{tab::single_col}  
  \begin{tabular}{|m{0.9in}|m{0.9in}|m{0.9in}|}  
    \hline  
    & 时间（秒） & 距离 \\\hline  
    A & 83.6 & 354 \\\hline  
    B & 80.2 & 402 \\\hline  
  \end{tabular}  
\end{table}
```

### 4.7.2 单栏纵向表格（一）

当表格的行数较少，栏数较多时，可考虑将表格纵向展示。当表格的内容较少时，可将表格插入在行文之中。如下例所示：



Methods	Time (Seconds)	Cost
Method A	10	100
Method B	11	120
Method C	12	130
Method D	13	140
Method E	14	150
Method F	15	160
Method G	16	170
Method H	17	180
Method I	18	190
Method J	19	200

表 1: An example of vertical tables.

writing in computer science, and presents research experience and skills in manuscript preparation, revision, and submission. All examples in this article are selected from papers published in top international conferences or journals. Although there are a variety of approaches to organize a research paper, this article and examples focus primarily on demonstrating methods usually adopted by the author of this article in paper writing. We hope that these materials will be helpful to researchers who are working hard to establish their research career.

This article introduces preliminaries of paper number of subsets. For each subset, a local skyline writing in computer science, and presents research

图 18: 单栏纵向表格。

此例的源代码如下。为了能使表格纵向展示，这里引入了adjustbox包（\usepackage{adjustbox}）。然后，在table的环境中（\begin{table}和\end{table}之间），建立了一个adjustbox的环境。angle=90和center是adjustbox环境的两个参数，分别说明该内容需要逆时针旋转90度，并在旋转后，居中对齐。在adjustbox环境中则是一个普通的表格内容。

#### 源代码示例

```
...
\usepackage{adjustbox}
...
\begin{table}[t]
  \begin{adjustbox}{angle=90, center}
    \centering
    \begin{tabular}{ccc}\hline
      Methods & Time (Seconds) & Cost \\\hline \hline
      Method A & 10 & 100 \\\hline
      Method B & 11 & 120 \\\hline
      Method C & 12 & 130 \\\hline
      Method D & 13 & 140 \\\hline
      Method E & 14 & 150 \\\hline
      Method E & 14 & 150 \\\hline
      Method F & 15 & 160 \\\hline
      Method G & 16 & 170 \\\hline
      Method H & 17 & 180 \\\hline
      Method I & 18 & 190 \\\hline
      Method J & 19 & 200 \\\hline \hline
    \end{tabular}
  \end{adjustbox}
\caption{An example of vertical tables.}
```



```
\end{table}
```

#### 4.7.3 单栏纵向表格（二）

如果纵向表格较大时，需要占满整个单栏（或者双栏：即占满整个页面），那么，可以使用rotating包。如下例所示：





experience and skills in manuscript preparation, revision, and submission. All examples in this article are selected from papers published in top international conferences or journals. Although there are a variety of approaches to organize a research paper, this article and examples focus primarily on demonstrating methods usually adopted by the author of this article in paper writing. We hope that these materials will be helpful to researchers who are working hard to establish their research career.

This article introduces preliminaries of paper writing in computer science, and presents research experience and skills in manuscript preparation, revision, and submission. All examples in this article are selected from papers published in top international conferences or journals. Although there are a variety of approaches to organize a research paper, this article and examples focus primarily on demonstrating methods usually adopted by the author of this article in paper writing. We hope that these materials will be helpful to researchers who are working hard to establish their research career.

This article introduces preliminaries of paper writing in computer science, and presents research experience and skills in manuscript preparation, revision, and submission. All examples in this article are selected from papers published in top international conferences or journals. Although there are a variety of approaches to organize a research paper, this article and examples focus primarily on demonstrating methods usually adopted by the author of this article in paper writing. We hope that these materials will be helpful to researchers who are working hard to establish their research career.

This article introduces preliminaries of paper writing in computer science, and presents research experience and skills in manuscript preparation, revision, and submission. All examples in this article are selected from papers published in top international conferences or journals. Although there are a variety of approaches to organize a research paper, this article and examples focus primarily on demonstrating methods usually adopted by the author of this article in paper writing. We hope that these materials will

Methods	Time (Seconds)	Cost	Extra Penalty
Method A	10	100	1100
Method B	11	120	2100
Method C	12	130	3100
Method D	13	140	4100
Method E	14	150	5100
Method F	15	160	6100
Method G	16	170	7100
Method H	17	180	8100
Method I	18	190	9100
Method J	19	200	1100

表 1: An example of vertical tables.

图 19: 单栏纵向表格。

此例使用了rotating包 (`\usepackage {rotating}`), 并在表格内容的外部建立了sidewaystable环境 (`\begin{sidewaystable}`和`\end{sidewaystable}`之间)。

#### 源代码示例

```
...
\usepackage{rotating}
...
\begin{sidewaystable}
\centering
\begin{tabular}{ccc}\hline
Methods & Time (Seconds) & Cost \\\hline
Method A & 10 & 100 \\\hline
```



```

Method B & 11 & 120 \\\hline
Method C & 12 & 130 \\\hline
Method D & 13 & 140 \\\hline
Method E & 14 & 150 \\\hline
Method E & 14 & 150 \\\hline
Method F & 15 & 160 \\\hline
Method G & 16 & 170 \\\hline
Method H & 17 & 180 \\\hline
Method I & 18 & 190 \\\hline
Method J & 19 & 200 \\\hline \hline
\end{tabular}
\caption{An example of vertical tables.}
\end{sidewaystable}

```

#### 4.7.4 跨栏表格

在双栏格式的论文中，因为单栏表格受到了单栏宽度的限制，在某些情况下，无法将所有数据放在一个表格中。此时，可使用跨栏表格以增加显示的宽度。跨栏表格会使用论文的整体宽度，因此，可以容纳更多的栏数。如下例所示。跨栏表使用`\begin{table*}`和`\end{table*}`环境指令。

表 1: Method Comparison.

	时间（秒）	距离
Method A	0/6/6	0/11/11
Method B	40/4/44	44/8/52
Method C	4/5/5	11/9/9

#### 源代码示例

```

\begin{table*}[h]
\centering
\caption{Comparison of skyline algorithms.}
\begin{tabular}{|>\centering\arraybackslash m{1.5in}|
>\centering\arraybackslash m{1.5in}||
>\centering\arraybackslash m{1.5in}|c|c|c|}
\hline
& 时间（秒） & 距离 \\\hline\hline
Method A & 0/6/6 & 0/11/11 \\\hline
Method B & 40/4/44 & 44/8/52 \\\hline
Method C & 4/5/5 & 11/9/9 \\\hline
\end{tabular}
\end{table*}

```



#### 4.7.5 合并单元格

在表格中，如果需要合并单元格，可使用`multirow`包和`multicolumn`包。从字面上可以得知，`multirow`包用于纵向合并单元格，`multicolumn`包用于横向合并单元格。

表 2: An example of merging cells.

Methods	DBs	Partitions				Total
		1	2	3	4	
A	DB1	310	3	7	169	489
	DB2	9	3	3	6	21
	DB3	835	658	651	836	2,980

#### 源代码示例

```
\begin{table}[h!]  
  \centering  
  \caption{An example of merging cells.}  
  \begin{tabular}{|c|c|c|c|c|c|c|}  
    \hline  
    \multirow{2}{*}{Methods} & \multirow{2}{*}{DBs} & & & & & \multirow{2}{*}{Total} \\ \hline  
    & & 1 & 2 & 3 & 4 & \\ \hline  
    \multirow{3}{*}{A} & DB1 & 310 & 3 & 7 & 169 & 489 \\ \hline  
    & DB2 & 9 & 3 & 3 & 6 & 21 \\ \hline  
    & DB3 & 835 & 658 & 651 & 836 & 2,980 \\ \hline  
  \end{tabular}  
\end{table}
```

## 4.8 数学模式与公式

### 4.8.1 数学模式

LaTeX支持两种模式。除了上述介绍的常用的文本编辑模式以外，LaTeX还支持数学模式。夹在两个美元符号之间( $\dots$ )的内容处于数学模式之中。例如，在正文中 $x_i$ 可以使用 $\$x_i\$$ 表达， $k^j$ 可以使用 $\$k^j\$$ 表达。

如果使用了`amsmath`包， $\[$ 是 $\begin{displaymath}$ 的缩写，它们能表达不带序号的公式。带序号的公式需使用 $\begin{equation}$ 和 $\end{equation}$ 环境。

一般来说，使用 $\dots$ 和 $\begin{equation}$ 环境足以应对论文中的各种情况。



### 4.8.2 公式

因为在数学模式中和在`equation`环境中，生成数学公式的使用方法是类似的。所以，下面仅展示数学公式的例子。在附录中列出了更多的数学符号，供参考使用。

#### 1. 上标和下标

$$x_i + y^j > z_i^j \quad (1)$$

##### 源代码示例

```
\begin{equation}
  x_i + y^j > z_i^j
\end{equation}
```

#### 2. 三角函数

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

##### 源代码示例

```
\begin{equation}
  \cos (2\theta) = \cos^2 \theta - \sin^2 \theta
\end{equation}
```

#### 3. 取模

$$a \bmod b \quad (3)$$

##### 源代码示例

```
\begin{equation}
  a \bmod b
\end{equation}
```

#### 4. 恒等

$$x \equiv a \pmod{b} \quad (4)$$

##### 源代码示例

```
\begin{equation}
```



```
x \equiv a \pmod{b}
\end{equation}
```

## 5. 幂

$$k_{n+1} = n^2 + k_n^2 - k_{n-1} \quad (5)$$

源代码示例

```
\begin{equation}
k_{n+1} = n^2 + k_n^2 - k_{n-1}
\end{equation}
```

## 6. 极限

$$\lim_{x \rightarrow \infty} \exp(-x) = 0 \quad (6)$$

源代码示例

```
\begin{equation}
\lim_{x \rightarrow \infty} \exp(-x) = 0
\end{equation}
```

## 7. 集合

$$\forall x \in X, \quad \exists y \leq \epsilon \quad (7)$$

源代码示例

```
\begin{equation}
\forall x \in X, \quad \exists y \leq \epsilon
\end{equation}
```

## 8. 分子与分母

$$\frac{n!}{k!(n-k)!} = \binom{n}{k} \frac{\frac{1}{x} + \frac{1}{y}}{y-z} 3 \times \frac{1}{2} = 1 \frac{1}{2} \quad (8)$$

源代码示例

```
\begin{equation}
\frac{n!}{k!(n-k)!} = \binom{n}{k} \frac{\frac{1}{x} + \frac{1}{y}}{y-z} \backslash \backslash
\end{equation}
```



```
3\times\frac{1}{2}=1\frac{1}{2}
\end{equation}
```

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4}}}} \quad (9)$$

#### 源代码示例

```
\begin{equation}
x = a_0 + \cfrac{1}{a_1}
+ \cfrac{1}{a_2}
+ \cfrac{1}{a_3 + \cfrac{1}{a_4} } }
\end{equation}
```

### 9. 乘积的计算

$$\frac{(x_1 x_2) \times (x'_1 x'_2)}{(y_1 y_2 y_3 y_4)} \quad (10)$$

#### 源代码示例

```
\begin{equation}
\frac{
\begin{array}{b}{r}
\left( x_1 x_2 \right) \backslash \backslash
\times \left( x'_1 x'_2 \right)
\end{array}
}{
\left( y_1 y_2 y_3 y_4 \right)
}
\end{equation}
```

### 10. 开根号

$$\sqrt[n]{\frac{a}{b} (1 + x + x^2 + x^3 + \dots + x^n)} \quad (11)$$



#### 源代码示例

```
\begin{equation}
\sqrt{\frac{a}{b}} \sqrt[n]{1+x+x^2+x^3+\dots+x^n}
\end{equation}
```

### 11. 求和

$$\sum_{i=1}^{10} t_i \sum_{i=1}^{10} t_i \sum_{\substack{0 \leq i < m \\ 0 \leq j < n}} P(i, j) \quad (12)$$

#### 源代码示例

```
\begin{equation}
\sum_{i=1}^{10} t_i \\\
\displaystyle \sum_{i=1}^{10} t_i \\\
\sum_{\substack{0 \leq i < m \\ 0 \leq j < n}} \\
P(i, j)
\end{equation}
```

### 12. 求导

$$\frac{d}{dx} (kg(x)) \left. \frac{x^3}{3} \right|_0^1 \quad (13)$$

#### 源代码示例

```
\begin{equation}
\frac{\mathrm{d}}{\mathrm{d} x} \left( k g(x) \right) \\\
\left. \frac{x^3}{3} \right|_0^1
\end{equation}
```

### 13. 积分

$$\int_0^{\infty} e^{-x} dx \quad (14)$$

#### 源代码示例

```
\begin{equation}
```





```
\int_0^\infty \mathrm{e}^{-x}\,,\mathrm{d}x
\end{equation}
```

#### 14. 概率

$$P\left(A=2\left|\frac{A^2}{B}>4\right.\right) \quad (15)$$

##### 源代码示例

```
\begin{equation}
P\left(A=2\middle|\frac{A^2}{B}>4\right)
\end{equation}
```

#### 15. 矩阵和数组

$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix} \quad (16)$$

##### 源代码示例

```
\begin{equation}
M = \begin{bmatrix}
\frac{5}{6} & \frac{1}{6} & 0 \\
\frac{5}{6} & 0 & \frac{1}{6} \\
0 & \frac{5}{6} & \frac{1}{6}
\end{bmatrix}
\end{equation}
```

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \quad (17)$$

##### 源代码示例

```
\begin{equation}
A_{\{m,n\}} =
\begin{pmatrix}
a_{\{1,1\}} & a_{\{1,2\}} & \cdots & a_{\{1,n\}} \\
a_{\{2,1\}} & a_{\{2,2\}} & \cdots & a_{\{2,n\}} \\
\vdots & \vdots & \ddots & \vdots
\end{pmatrix}
\end{equation}
```



```
a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{pmatrix}
\end{equation}
```

## 16. 分段函数

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ -(n+1)/2 & \text{if } n \text{ is odd} \end{cases}$$

### 源代码示例

```
\[ f(n) =
\begin{cases}
n/2 & \text{if } n \text{ is even} \\
-(n+1)/2 & \text{if } n \text{ is odd}
\end{cases}
\]
```

## 17. 括号的使用

$$(a), [b], \{c\}, |d|, \|e\|, \langle f \rangle, \lfloor g \rfloor, \lceil h \rceil, \lceil i \rceil \quad (18)$$

### 源代码示例

```
\begin{equation}
(a), [b], \{c\}, |d|, \|e\|,
\lceil f \rceil, \lfloor g \rfloor,
\lceil h \rceil, \lfloor i \rfloor
\end{equation}
```

## 4.9 算法

算法是论文中重要的组成部分之一。本文使用algorithm和algorithm2e包展示了if-else分支，for分支和while分支的例子以及源代码。LaTeX环境下还有其他的算法包，它们的使用方法类似，可参考相关的文档。

算法只能单栏显示，其长度尽量不要超过页面的长度，否则，会造成一些论文排版的麻烦。算法一般由序号，标题和算法主体组成。算法序号用于在论文中唯一标识某算法。算法标题说明了算法的用途。算法主体则是表达了算法的逻辑。



#### 4.9.1 algorithm包

##### IF-ELSE条件分支

---

**Algorithm 1**  $\max(a, b)$ 

---

```
1: if  $a > b$  then  
2:   return  $a$ ;  
3: else  
4:   return  $b$ ;  
5: end if
```

---

##### 源代码示例

```
\begin{algorithm}[h!]  
  \caption{\mathit{\max}(a, b)}  
  \begin{algorithmic}[1]  
    \If { $a > b$ }  
      \State \Return  $a$ ;  
    \Else  
      \State \Return  $b$ ;  
    \EndIf  
  \end{algorithmic}  
\end{algorithm}
```

##### FOR循环

---

**Algorithm 2**  $\max(Q)$ 

---

```
1:  $m = Q[0]$ ;  
2: for  $q \in Q$  do  
3:   if  $m < q$  then  
4:      $m = q$ ;  
5:   else  
6:     Continue;  
7:   end if  
8: end for  
9: return  $m$ ;
```

---

##### 源代码示例

```
\begin{algorithm}[h!]  
  \caption{\mathit{\max}(Q)}  
  \begin{algorithmic}[1]  
    \State  $m = Q[0]$ ;  
    \For {  $q \in Q$  }  
      \If {  $m < q$  }  
        \State  $m = q$ ;  
      \EndIf  
    \EndFor  
  \end{algorithmic}  
\end{algorithm}
```



```

\Else
\State Continue;
\EndIf
\EndFor
\State \Return  $m$ ;
\end{algorithmic}
\end{algorithm}

```

## WHILE循环

---

### Algorithm 3 $\max(Q)$

---

```

1:  $m = Q[0]$ ;
2:  $i = \text{getIterator}(Q)$ ;
3: while  $i.\text{hasNext}()$  do
4:    $t = i.\text{next}()$ ;
5:   if  $m < t$  then
6:      $m = t$ ;
7:   end if
8: end while
9: return  $m$ ;

```

---

#### 源代码示例

```

\begin{algorithm}[h!]
\caption{\mathmax(Q)}
\begin{algorithmic}[1]
\State  $m = Q[0]$ ;
\State  $i = \text{getIterator}(Q)$ ;
\While { $i.\text{hasNext}()$ }
\State  $t = i.\text{next}()$ ;
\If { $m < t$ }
\State  $m = t$ ;
\EndIf
\EndWhile
\State \Return  $m$ ;
\end{algorithmic}
\end{algorithm}

```

## 4.9.2 algorithm2e包

### IF-ELSE条件分支



---

**Algorithm 1:** MAX finds the greater integer

---

**Input:** Two integers  $a$  and  $b$

**Output:** The greater integer

```
1 if  $a > b$  then
2   return  $a$ 
3 else
4   return  $b$ 
```

---

源代码示例

```
\begin{algorithm}
  \DontPrintSemicolon
  \KwIn{Two integers  $a$  and  $b$ }
  \KwOut{The greater integer}
  \If{ $a > b$ } {
    \Return{ $a$ }
  }
  \Else{
    \Return{ $b$ }
  }
  \caption{{\sc Max} finds the greater number}
\end{algorithm}
```

FOR循环

---

**Algorithm 1:** MAX finds the greatest integer

---

**Input:** A set  $Q = \{q_1, \dots, q_n\}$  of integers

**Output:** The greatest integer in the set

```
1  $m \leftarrow Q[0]$ 
2 for  $q \in Q$  do
3   if  $m < q$  then
4      $m \leftarrow q$ 
5   else
6     Continue
7 return  $m$ 
```

---

源代码示例

```
\begin{algorithm}
  \DontPrintSemicolon
  \KwIn{A set  $Q = \{q_1, \dots, q_n\}$  of integers}
  \KwOut{The greatest integer}
```



```

 $m \leftarrow Q[0]$ 
\For $\{q \in Q\}$  {
  \If $\{m < q\}$  {
     $m \leftarrow q$ 
  }
  \Else{
    Continue
  }
}
\Return $m$ 
\caption{{\sc Max} finds the greater number}
\end{algorithm}

```

## WHILE循环

---

### Algorithm 1: MAX finds the greater number

---

**Input:** A set  $Q = \{q_1, \dots, q_n\}$  of integers

**Output:** The greatest integer

```

1  $m \leftarrow Q[0]$ 
2 while  $q \in Q$  do
3   if  $m < q$  then
4      $m \leftarrow q$ 
5   else
6     Continue
7 return  $m$ 

```

---

### 源代码示例

```

\begin{algorithm}
\ DontPrintSemicolon
\ KwIn{A set  $Q = \{q_1, \dots, q_n\}$  of integers}
\ KwOut{The greatest integer}
 $m \leftarrow Q[0]$ 
\ While $\{q \in Q\}$  {
  \ If $\{m < q\}$  {
     $m \leftarrow q$ 
  }
  \ Else{
    Continue
  }
}
\ Return $m$ 

```



```
\caption{{\sc Max} finds the greater number}  
\end{algorithm}
```

## 4.10 脚注

当论文正文内容需要更详细的解释，但是在正文直接给出可能会打断阅读的思路，所以，此时可考虑将解释的内容放在脚注中。使用`\footnote{}`命令标识脚注的内容，如下图所示。

amples in this article are selected from papers published in top conferences or journals. Although there are a variety of approaches to organize a research paper, this article and examples focus primarily on demonstrating methods usually adopted by the author of this article in paper writing. We hope

top conferences or journals. Although there are a variety of approaches to organize a research paper, this article and examples focus primarily on demonstrating methods usually

<sup>1</sup>This is an example of footnote.

图 20: 脚注的例子。





## 4.11 文献

通常情况下，在论文的结尾处使用`\bibliographystyle{}`和`\bibliography{}`命令引入参考文献列表。最常用的`\bibliographystyle`命令的参数为`abbrv`和`plain`。`plain`指列出作者全名和会议或者期刊的全名。`abbrv`则指使用作者名，会议名和期刊名的缩写。`\bibliography{}`命令接受一个文件名，该文件包含了所有该论文引用的参考文献。这个文件常以`.bib`为后缀名。在使用`\bibliography{}`命令时，传入的文件名参数不包含后缀。

以下列举了在`.bib`文件中论文，书籍，网页等参考文件的格式。

### 论文

#### 源代码示例

```
@article{DBLP:journals/tpds/ZhangJKQ16,  
  author = {Ji Zhang and  
    Xunfei Jiang and  
    Wei{-}Shinn Ku and  
    Xiao Qin},  
  title = {{Efficient Parallel Skyline Evaluation Using MapReduce}},  
  journal = {{IEEE} Trans. Parallel Distrib. Syst.},  
  volume = {27},  
  number = {7},  
  pages = {1996-2009},  
  year = {2016}  
}
```

#### 源代码示例

```
@inproceedings{DBLP:conf/edbt/WangZSK17,  
  author = {Wenlu Wang and  
    Ji Zhang and  
    Min{-}Te Sun and  
    Wei{-}Shinn Ku},  
  title = {{Efficient Parallel Spatial Skyline Evaluation Using MapReduce}},  
  booktitle = {Proceedings of the 20th International Conference on Extending  
    Database Technology, {EDBT} 2017, Venice, Italy, March 21-24, 2017.},  
  pages = {426-437},  
  year = {2017}  
}
```



#### 源代码示例

```
@proceedings{DBLP:conf/edbt/2017,  
  editor = {Volker Markl and  
    Salvatore Orlando and  
    Bernhard Mitschang and  
    Periklis Andritsos and  
    Kai{-}Uwe Sattler and  
    Sebastian Bre{ß}},  
  title = {Proceedings of the 20th International Conference on Extending Database  
    Technology, {EDBT} 2017, Venice, Italy, March 21-24, 2017},  
  publisher = {OpenProceedings.org},  
  year = {2017}  
}
```

#### 书籍

#### 源代码示例

```
@book{DBLP:reference/sp/2015dc,  
  editor = {Samee U. Khan and  
    Albert Y. Zomaya},  
  title = {Handbook on Data Centers},  
  publisher = {Springer},  
  year = {2015}  
}
```

#### 网页

#### 源代码示例

```
@misc{knuthwebsite,  
  author = "Little Waterdrop",  
  title = "Introduction to Research Papers in Computer Science",  
  url = "http://www.littlewaterdrop.com"  
}
```



## 4.12 符号表(Symbol Lists)

### 4.12.1 希腊字母表(Greek Alphabet)

表 3: 希腊字母表 (数学模式下)

$\alpha$	<code>\alpha</code>	$\beta$	<code>\beta</code>
$\gamma$ $\Gamma$	<code>\gamma</code> <code>\Gamma</code>	$\delta$ $\Delta$	<code>\delta</code> <code>\Delta</code>
$\epsilon$ $\varepsilon$	<code>\epsilon</code> <code>\varepsilon</code>	$\zeta$	<code>\zeta</code>
$\eta$	<code>\eta</code>	$\theta$ $\vartheta$ $\Theta$	<code>\theta</code> <code>\vartheta</code> <code>\Theta</code>
$\iota$	<code>\iota</code>	$\kappa$ $\varkappa$	<code>\kappa</code> <code>\varkappa</code>
$\lambda$ $\Lambda$	<code>\lambda</code> <code>\Lambda</code>	$\mu$	<code>\mu</code>
$\nu$	<code>\nu</code>	$\xi$ $\Xi$	<code>\xi</code> <code>\Xi</code>
$\pi$ $\Pi$	<code>\pi</code> <code>\Pi</code>	$\rho$ $\varrho$	<code>\rho</code> <code>\varrho</code>
$\sigma$ $\Sigma$	<code>\sigma</code> <code>\Sigma</code>	$\tau$	<code>\tau</code>
$\upsilon$ $\Upsilon$	<code>\upsilon</code> <code>\Upsilon</code>	$\phi$ $\varphi$ $\Phi$	<code>\phi</code> <code>\varphi</code> <code>\Phi</code>
$\chi$	<code>\chi</code>	$\psi$ $\Psi$	<code>\psi</code> <code>\Psi</code>
$\omega$ $\Omega$	<code>\omega</code> <code>\Omega</code>	$\digamma$	<code>\digamma</code>
$\aleph$	<code>\aleph</code>	$\beth$	<code>\beth</code>
$\daleth$	<code>\daleth</code>	$\gimel$	<code>\gimel</code>



#### 4.12.2 函数名表(Function List)

表 4: 函数名表 (数学模式下)

exp	\exp	ker	\ker
lim sup	\limsup	min	\min
deg	\deg	gcd	\gcd
lg	\lg	ln	\ln
Pr	\Pr	sup	\sup
det	\det	hom	\hom
lim	\lim	log	\log
sec	\sec	arg	\arg
dim	\dim	inf	\inf
lim inf	\liminf	max	\max

表 5: 三角函数表 (数学模式下)

sin	\sin	cos	\cos
tan	\tan	arcsin	\arcsin
arccos	\arccos	arctan	\arctan
sinh	\sinh	cosh	\cosh
tanh	\tanh	coth	\coth
sec	\sec	csc	\csc



### 4.12.3 数学符号表(Mathematical Symbols)

表 6: 数学符号表 (数学模式下)

$\frac{x}{y}$	<code>\frac{x}{y}</code>	$f'$	<code>f'</code>
$\sqrt{x}$	<code>\sqrt{x}</code>	$\sqrt[n]{x}$	<code>\sqrt[n]{x}</code>
$\overline{x}$	<code>\overline{x}</code>	$\underline{x}$	<code>\underline{x}</code>
$\widehat{x}$	<code>\widehat{x}</code>	$\widetilde{x}$	<code>\widetilde{x}</code>
$\overrightarrow{x}$	<code>\overrightarrow{x}</code>	$\overleftarrow{x}$	<code>\overleftarrow{x}</code>
$\overbrace{x}$	<code>\overbrace{x}</code>	$\underbrace{x}$	<code>\underbrace{x}</code>
$\sum x$	<code>\sum{x}</code>	$\prod x$	<code>\prod{x}</code>
$\coprod x$	<code>\coprod{x}</code>	$\int x$	<code>\int{x}</code>
$\oint x$	<code>\oint{x}</code>	$\iint x$	<code>\iint{x}</code>
$\biguplus x$	<code>\biguplus{x}</code>	$\bigcap x$	<code>\bigcap{x}</code>
$\bigcup x$	<code>\bigcup{x}</code>	$\bigoplus x$	<code>\bigoplus{x}</code>
$\bigotimes x$	<code>\bigotimes{x}</code>	$\bigodot x$	<code>\bigodot{x}</code>
$\bigvee x$	<code>\bigvee{x}</code>	$\bigwedge x$	<code>\bigwedge{x}</code>
$\bigsqcup x$	<code>\bigsqcup{x}</code>		
$\infty$	<code>\infty</code>	$\forall$	<code>\forall</code>
$\exists$	<code>\exists</code>	$\partial$	<code>\partial</code>
$\nexists$	<code>\nexists</code>	$\emptyset$	<code>\emptyset</code>
$\varnothing$	<code>\varnothing</code>	$\ell$	<code>\ell</code>
$\cdots$	<code>\cdots</code>	$\sharp$	<code>\sharp</code>
$\Im$	<code>\Im</code>	$\Re$	<code>\Re</code>
$\diagdown$	<code>\diagdown</code>	$\diagup$	<code>\diagup</code>
$\Diamond$	<code>\Diamond</code>	$'$	<code>\prime</code>
$\hbar$	<code>\hbar</code>	$\hslash$	<code>\hslash</code>
$\square$	<code>\square</code>	$\surd$	<code>\surd</code>
$\angle$	<code>\angle</code>	$\measuredangle$	<code>\measuredangle</code>
$\sphericalangle$	<code>\sphericalangle</code>	$\complement$	<code>\complement</code>
$\triangledown$	<code>\triangledown</code>	$\triangle$	<code>\triangle</code>



#### 4.12.4 操作符表(Operator List)

表 7: 操作符表 (数学模式下)

*	\ast	★	\star
·	\cdot	○	\circ
●	\bullet	◯	\bigcirc
◇	\diamond	×	\times
÷	\div	•	\centerdot
⊗	\circledast	⊙	\circledcirc
⊖	\circleddash	+	\dotplus
÷	\divideontimes	±	\pm
⋈	\mp	Π	\amalg
⊙	\odot	⊖	\ominus
⊕	\oplus	⊗	\oslash
⊗	\otimes	ℳ	\wr
□	\Box	⊞	\boxplus
⊞	\boxminus	⊠	\boxtimes
⊠	\boxdot	□	\square
∩	\cap	∪	\cup
⊕	\uplus	∩	\sqcap
∪	\sqcup	∧	\wedge
∨	\vee	†	\dagger
‡	\ddagger	⋈	\barwedge
⋈	\curlywedge	⌞	\Cap
⊥	\bot	⌞	\intercal
⋈	\doublebarwedge	◁	\lhd
▷	\rhd	◁	\triangleleft
▷	\triangleright	◁	\unlhd
▷	\unrhd	▽	\bigtriangledown
△	\bigtriangleup	\	\setminus
∨	\veebar	Υ	\curlyvee
⊞	\Cup	⊤	\top
×	\rightthreetimes	λ	\leftthreetimes



表 8: 操作符表（数学模式下）续

$\equiv$	<code>\equiv</code>	$\cong$	<code>\cong</code>
$\neq$	<code>\neq</code>	$\sim$	<code>\sim</code>
$\simeq$	<code>\simeq</code>	$\approx$	<code>\approx</code>
$\asymp$	<code>\asymp</code>	$\doteq$	<code>\doteq</code>
$\propto$	<code>\propto</code>	$\models$	<code>\models</code>
$\leq$	<code>\leq</code>	$\prec$	<code>\prec</code>
$\preceq$	<code>\preceq</code>	$\ll$	<code>\ll</code>
$\subset$	<code>\subset</code>	$\subseteq$	<code>\subseteq</code>
$\sqsubset$	<code>\sqsubset</code>	$\sqsubseteq$	<code>\sqsubseteq</code>
$\dashv$	<code>\dashv</code>	$\in$	<code>\in</code>
$\geq$	<code>\geq</code>	$\succ$	<code>\succ</code>
$\succeq$	<code>\succeq</code>	$\gg$	<code>\gg</code>
$\supset$	<code>\supset</code>	$\supseteq$	<code>\supseteq</code>
$\sqsupset$	<code>\sqsupset</code>	$\sqsupseteq$	<code>\sqsupseteq</code>
$\vdash$	<code>\vdash</code>	$\ni$	<code>\ni</code>
$\perp$	<code>\perp</code>	$\mid$	<code>\mid</code>
$\parallel$	<code>\parallel</code>	$\bowtie$	<code>\bowtie</code>
$\Join$	<code>\Join</code>	$\ltimes$	<code>\ltimes</code>
$\rtimes$	<code>\rtimes</code>	$\notin$	<code>\notin</code>
$\approx$	<code>\approx</code>	$\thicksim$	<code>\thicksim</code>
$\backsim$	<code>\backsim</code>	$\backsimeq$	<code>\backsimeq</code>
$\triangleq$	<code>\triangleq</code>	$\circeq$	<code>\circeq</code>
$\varpropto$	<code>\varpropto</code>	$\therefore$	<code>\therefore</code>
$\because$	<code>\because</code>	$\eqcirc$	<code>\eqcirc</code>
$\neq$	<code>\neq</code>	$\leqq$	<code>\leqq</code>
$\leqslant$	<code>\leqslant</code>	$\lessapprox$	<code>\lessapprox</code>
$\lll$	<code>\lll</code>	$\lessdot$	<code>\lessdot</code>
$\lesssim$	<code>\lesssim</code>	$\eqslantless$	<code>\eqslantless</code>
$\prec$	<code>\prec</code>	$\precapprox$	<code>\precapprox</code>
$\Subset$	<code>\Subset</code>	$\subseteqq$	<code>\subseteqq</code>
$\preccurlyeq$	<code>\preccurlyeq</code>	$\curlyeqprec$	<code>\curlyeqprec</code>
$\geqq$	<code>\geqq</code>	$\trianglelefteq$	<code>\trianglelefteq</code>
$\geqslant$	<code>\geqslant</code>	$\gtrapprox$	<code>\gtrapprox</code>





表 9: 操作符表（数学模式下）续

$\gg$	<code>\ggg</code>	$\triangleright$	<code>\gtrdot</code>
$\gtrsim$	<code>\gtrsim</code>	$\triangleright$	<code>\eqslantgtr</code>
$\succsim$	<code>\succsim</code>	$\succsim$	<code>\succapprox</code>
$\supset$	<code>\Supset</code>	$\supset$	<code>\supseteqq</code>
$\sqsupset$	<code>\sqsupset</code>	$\succcurlyeq$	<code>\succcurlyeq</code>
$\curlyeqsucc$	<code>\curlyeqsucc</code>	$\blacktriangleright$	<code>\blacktriangleright</code>
$\trianglerighteq$	<code>\trianglerighteq</code>	$\triangleright$	<code>\vartriangleright</code>
$\lessgtr$	<code>\lessgtr</code>	$\lessgtr$	<code>\lesseqgtr</code>
$\lesseqqgtr$	<code>\lesseqqgtr</code>	$\gtreqless$	<code>\gtreqless</code>
$\gtreqless$	<code>\gtreqless</code>	$\gtrless$	<code>\gtrless</code>
$\backepsilon$	<code>\backepsilon</code>	$\between$	<code>\between</code>
$\vartriangleleft$	<code>\vartriangleleft</code>	$\blacktriangleleft$	<code>\blacktriangleleft</code>
$\nleq$	<code>\nleq</code>	$\nleqq$	<code>\nleqq</code>
$\nleqslant$	<code>\nleqslant</code>	$\nless$	<code>\nless</code>
$\npreceq$	<code>\npreceq</code>	$\precapprox$	<code>\precapprox</code>
$\lneq$	<code>\lneq</code>	$\lneqq$	<code>\lneqq</code>
$\lnsim$	<code>\lnsim</code>	$\lvertneqq$	<code>\lvertneqq</code>
$\ngeq$	<code>\ngeq</code>	$\ngeqq$	<code>\ngeqq</code>
$\ngeqslant$	<code>\ngeqslant</code>	$\ngtr$	<code>\ngtr</code>
$\nsucc$	<code>\nsucc</code>	$\nsucceq$	<code>\nsucceq</code>
$\succapprox$	<code>\succapprox</code>	$\succsim$	<code>\succsim</code>
$\gneq$	<code>\gneq</code>	$\gneqq$	<code>\gneqq</code>
$\gnsim$	<code>\gnsim</code>	$\gvertneqq$	<code>\gvertneqq</code>
$\nsubseteq$	<code>\nsubseteq</code>	$\nsupseteq$	<code>\nsupseteq</code>
$\nsubseteqq$	<code>\nsubseteqq</code>	$\nsupseteqq$	<code>\nsupseteqq</code>
$\subsetneq$	<code>\subsetneq</code>	$\supsetneq$	<code>\supsetneq</code>
$\supsetneqq$	<code>\supsetneqq</code>	$\varsubsetneq$	<code>\varsubsetneq</code>
$\varsupsetneq$	<code>\varsupsetneq</code>	$\varsubsetneqq$	<code>\varsubsetneqq</code>
$\varsupsetneqq$	<code>\varsupsetneqq</code>		



#### 4.12.5 箭头符号表(Arrow Symbols)

表 10: 箭头符号表 (数学模式下)

$\leftarrow$	<code>\leftarrow</code>	$\Leftarrow$	<code>\Leftarrow</code>
$\rightarrow$	<code>\rightarrow</code>	$\Rightarrow$	<code>\Rightarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\Leftrightarrow$	<code>\Leftrightarrow</code>
$\longleftarrow$	<code>\longleftarrow</code>	$\Longleftarrow$	<code>\Longleftarrow</code>
$\longrightarrow$	<code>\longrightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>
$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\Longleftrightarrow$	<code>\Longleftrightarrow</code>
$\uparrow$	<code>\uparrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\downarrow$	<code>\downarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\updownarrow$	<code>\updownarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\leftharpoondown$	<code>\leftharpoondown</code>
$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\rightharpoonup$	<code>\rightharpoonup</code>
$\rightharpoondown$	<code>\rightharpoondown</code>	$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>
$\leftarrowtail$	<code>\leftarrowtail</code>	$\rightsquigarrow$	<code>\rightsquigarrow</code>
$\nleftarrow$	<code>\nleftarrow</code>	$\dashleftarrow$	<code>\dashleftarrow</code>
$\rightarrowtail$	<code>\rightarrowtail</code>	$\nrightarrow$	<code>\nrightarrow</code>
$\nrightarrow$	<code>\nrightarrow</code>	$\nLeftarrow$	<code>\nLeftarrow</code>
$\nRightarrow$	<code>\nRightarrow</code>	$\nleftrightarrow$	<code>\nleftrightarrow</code>
$\nLeftrightarrow$	<code>\nLeftrightarrow</code>		



#### 4.12.6 特殊字母表(Others)

表 11: 特殊字母表（文本模式下）

ó	\'o}	ö	\"o}
ô	\.o}	û	\b{o}
ø	\o	æ	\ae
ö	\u{o}	ô	\{o}
õ	\H{o}	ò	\'o}
öö	\t{oo}	õ	\{o}
ø	\c{o}	ō	\={o}
ø	\d{o}	ş	\d s
š	\r s	š	\H s
Å	\AA	å	\aa
ß	\ss	ı	\i
Ј	\j	š	\t s
š	\v s	Ø	\O
¶	\P	§	\S
Æ	\AE	†	\dag
‡	\ddag	©	\copyright
£	\pounds		

表 12: 特殊字母表（数学模式下）

á	\acute{a}	<i>brevea</i>	\breve{a}
ä	\ddot{a}	à	\grave{a}
ã	\tilde{a}	ā	\bar{a}
ǎ	\check{a}	ȁ	\dot{a}
â	\hat{a}	ā	\vec{a}
Á	\Acute{\Acute{A}}	Ă	\Breve{\Breve{A}}
Ä	\Ddot{\Ddot{A}}	À	\Grave{\Grave{A}}
Ã	\Tilde{\Tilde{A}}	Ā	\Bar{\Bar{A}}
Ǻ	\Check{\Check{A}}	Ȧ	\Dot{\Dot{A}}
Â	\Hat{\Hat{A}}	Ȧ	\Vec{\Vec{A}}



## 5 附录 (Appendix)

### 5.1 会议与期刊排名 (Conference and Journal Rankings)

#### 5.1.1 中国计算机学会(CCF)国际学术会议和期刊目录

中国计算机学会在其推荐的国际学术会议和期刊目录中列举了以下十个领域中的国际著名的学术会议和期刊。其中，A类指国际上极少数的顶级会议和刊物；B类为国际上著名和非常重要的会议和刊物，有重要的学术影响。会议论文是指的“Full Paper”或“Regular Paper”。其他形式的论文（例如Short Paper, Poster, Demo Paper等）不计入在内。可在如下地址查看其会议和期刊列表<https://www.ccf.org.cn/xspj/gym1/>。

#### 5.1.2 中文核心期刊目录

中文核心期刊目录是对于中文学术期刊等级的划分。该划分考量了期刊影响因子、被引量、被引量、他引量等诸多因素，且由众多学术界权威专家联合鉴定。中文核心期刊目录包含了各个学科的学术刊物。其中，自动化技术和计算机技术类别包含了30个核心学术期刊。

#### 5.1.3 Computing Research and Education Association of Australasia (CORE)

CORE是在澳大利亚和新西兰的一个计算科研与教育协会的简称。这个协会专门为计算机科学与信息技术领域的高等教育和科研机构提供服务。同时，这个协会也维护着一个计算机会议和期刊的排名。例如，A\*表示世界顶级的会议和期刊，A表示世界优秀的会议和期刊。在2018年会议排名中，只有4%的会议被列为A\*。在2010年的期刊排名中，只有7%的期刊被列为A\*。

#### 5.1.4 计算机会议和期刊排名（非官方）

在计算机学术界流传着一个计算机会议的排名名单。该名单不是任何机构或组织公布的正式名单。这份名单更像是学者心中的会议排名。

### 计算机期刊排名

#### 硬件和体系结构(Hardware and Architecture)

##### Rank 1

1 IEEE Transactions on Circuits and Systems I



- 2 IEEE Transactions on Computers (TOC)
- 3 Integration: The VLSI Journal

## Rank 2

- 1 IEEE Transactions on Circuits and Systems II
- 2 Journal of Microcomputer Applications
- 3 Microprocessing and Microprogramming

## 系统技术(System Technology)

### Rank 1

- 1 IEEE Transactions on Communications
- 2 Journal of Parallel and Distributed Computing
- 3 ACM Transactions on Computer Systems
- 4 IEEE/ACM Transactions on Networking
- 5 ACM Transactions on Information Systems
- 6 IEEE Transactions on Parallel and Distributed Systems
- 7 Distributed Computing
- 8 IEEE Journal on Selected Areas in Communication
- 9 International Journal of Parallel Programming
- 10 Computer Networks
- 11 Real-time Systems

### Rank 2

- 1 Networks
- 2 Internet Research - Electronic Networking Applications and Policy
- 3 Computer Communications
- 4 Journal of Computer Communications
- 5 Journal of Network and Computer Applications
- 6 Parallel Computing
- 7 Mobile Networks and Applications
- 8 Journal of High Speed Networks
- 9 Wireless Networks
- 10 Queueing Systems



- 11 The International Journal of Supercomputer Applications
- 12 Concurrency - Practice and Experience
- 13 IEEE Journal of Internet Computing
- 14 ACM Transactions on Information and System Security

## 编程语言和软件工程(Programming Languages and Software Engineering)

### Rank 1

- 1 ACM Transactions on Programming Languages and Systems
- 2 Annals of Software Engineering
- 3 IEEE Transactions on Software Engineering
- 4 Journal of Functional Programming
- 5 ACM Transactions on Software Engineering and Methodology
- 6 Formal Methods in System Design

### Rank 2

- 1 The Journal of Logic Programming
- 2 IEE Proceedings - Software
- 3 Journal of Software Maintenance: Research and Practice
- 4 Higher-Order and Symbolic Computation (previously known as LISP and Symbolic Computation)
- 5 Software: Practice and Experience
- 6 Journal of Functional and Logic Programming
- 7 The Constraints Journal
- 8 Journal of Logic and Computation
- 9 Journal of Programming Languages
- 10 Empirical Software Engineering
- 11 Automated Software Engineering
- 12 Formal Aspects of Computing
- 13 Object-Oriented Systems
- 14 Theory and Practice of Object Systems
- 15 Journal of Object-Oriented Programming
- 16 IEEE Transactions on Reliability
- 17 Future Generations Computer Systems: FGCS (Grid Computing)



## 数据库(Data Bases)

### Rank 1

- 1 ACM Transactions on Database Systems
- 2 IEEE Transactions on Knowledge and Data Engineering
- 3 Journal of Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies
- 4 VLDB Journal
- 5 Distributed and Parallel Databases

### Rank 2

- 1 Data and Knowledge Engineering
- 2 Information systems
- 3 Journal of Systems Integration
- 4 Journal of Data Mining and Knowledge Discovery
- 5 International Journal of Computer and Information Sciences
- 6 International Journal of Cooperative Information Systems
- 7 International Journal of Intelligent and Cooperative Information Systems
- 8 International Journal of Geographic Information Systems
- 9 Journal of Information Processing and Cybernetics
- 10 Geoinformatica
- 11 Journal on Digital Libraries
- 12 Journal of the American Society for Information Science
- 13 Journal of Intelligent Information Systems

## 算法, 理论及相关领域(Algorithms, Theory and Related Areas)

### Rank 1

- 1 Algorithmica
- 2 Computational Complexity
- 3 Discrete and Computational Geometry
- 4 IEEE Transactions on Information Theory
- 5 Information and Computation





- 6 Journal of Algorithms
- 7 Journal of Computer and System Sciences
- 8 Journal of the Association for Computing Machinery
- 9 SIAM Journal on Computing
- 10 Mathematics of OR

## Rank 2

- 1 Acta Informatica
- 2 Chicago Journal of Theoretical Computer Science
- 3 Computational Logic (TOCL)
- 4 Designs, Codes and Cryptography
- 5 Journal of Symbolic Computation
- 6 Journal of Automated Reasoning
- 7 Journal of Graph Algorithms and Applications
- 8 Journal of Complexity
- 9 Journal of Cryptology
- 10 JOTA - Journal of Optimization: Theory and Applications
- 11 Mathematics of Computation
- 12 Mathematical Programming
- 13 Optimization: A Journal of Mathematical Programming and Operations Research
- 14 ORSA Journal of Computing
- 15 Nordic Journal of Computing (BIT)
- 16 SIAM Journal on Optimization
- 17 Random Structures and Algorithms
- 18 Theoretical Computer Science

## 计算机图形与人机交互(Computer Graphics and Interaction)

### Rank 1

- 1 ACM Transactions on Graphics
- 2 ACM Transactions on Modeling and Computer Simulation
- 3 Computer Aided Geometric Design
- 4 Computer-Aided Design



- 5 IEEE Transactions on CAD of Integrated Circuits and Systems
- 6 IEEE Transactions on Visualization and Computer Graphics
- 7 SIAM Journal on Scientific and Statistical Computing
- 8 Multimedia Systems
- 9 Performance Evaluation
- 10 Journal of Visual Communication and Image Representation

## Rank 2

- 1 Computers and Education
- 2 ACM Transactions on Mathematical Software
- 3 Hypermedia
- 4 International Journal of Modeling and Simulation
- 5 International Journal of Shape Modeling
- 6 International Journal on Computational Geometry and Apps
- 7 Simulation and Games
- 8 Simulation and Gaming
- 9 Visual Computer
- 10 Computational Geometry - Theory and Applications
- 11 Simulation
- 12 Multimedia Tools and Applications
- 13 International Journal in Computer Simulation
- 14 Integrated Computer-Aided Engineering
- 15 Information Retrieval

## 人工智能(Artificial Intelligence)

### Rank 1

- 1 Artificial Intelligence
- 2 Artificial Intelligence Review
- 3 Computational Linguistics
- 4 IEEE Transactions on Pattern Analysis and Machine Intelligence
- 5 IEEE Transactions on Robotics and Automation
- 6 IEEE Transactions on Image Processing



- 7 Journal of Artificial Intelligence Research
- 8 Neural Computation
- 9 Machine Learning
- 10 International Journal of Computer Vision

## Rank 2

- 1 ACM Transactions on Asian Language Information Processing
- 2 Artificial Intelligence Magazine
- 3 Annals of Mathematics and Artificial Intelligence
- 4 Applied Artificial Intelligence
- 5 Applied Intelligence
- 6 Artificial Intelligence in Medicine
- 7 Autonomous Agents and Multi-Agent Systems
- 8 Computational Intelligence
- 9 Complex Systems
- 10 Computer Speech and Language
- 11 Computer Support for Collaborative Learning (CSCL)
- 12 Computer Vision and Image Understanding
- 13 Connection Science
- 14 CVGIP: Graphical Models and Image Processing
- 15 CVGIP: Image Understanding
- 16 Expert Systems with Applications: Artificial Intelligence International Journal
- 17 IEEE Transactions on Neural Networks
- 18 IEEE Transactions on Speech and Audio Processing
- 19 IEEE Transactions on Systems, Man, and Cybernetics
- 20 International Journal on Artificial Intelligence Tools
- 21 Journal of Experimental and Theoretical Artificial Intelligence
- 22 Journal of East Asian Linguistics
- 23 Knowledge Engineering Review
- 24 Machine Translation
- 25 Neural Networks
- 26 Network Computing in Neural Systems
- 27 Pattern Analysis and Applications



- 28 Pattern Recognition
- 29 Neurocomputing
- 30 User Modeling and User-Adapted Interaction: an International Journal

## 通用计算机科学及相关领域(General Computer Science and Related Journals)

### Rank 1

- 1 Journal of the American Medical Informatics Assoc (JAMIA)
- 2 Journal of Combinatorial Theory
- 3 Combinatorica
- 4 SIAM Journal on Discrete Mathematics
- 5 Operations Research

### Rank 2

- 1 Communications of the ACM
- 2 ACM Computing Surveys
- 3 Proceedings of the IEEE
- 4 Information Processing Letters
- 5 IBM Journal of Research and Development
- 6 Computers in Biomedical Research
- 7 Methods of Information in Medicine
- 8 Medical Decision Making
- 9 International Journal of General Systems
- 10 Adaptive Behavior
- 11 Discrete Applied Mathematics
- 12 Discrete Mathematics
- 13 Journal of Computational Biology
- 14 Journal of Molecular Biology
- 15 Bioinformatics

## 计算机会议排名

### 硬件与体系结构(Hardware and Architecture)



## Rank 1

- 1 ASPLOS: ACM Architectural Support for Programming Languages and Operating Systems
- 2 ISCA: ACM/IEEE Symposium on Computer Architecture
- 3 ICCAD: International Conference on Computer-Aided Design
- 4 DAC: Design Automation Conference
- 5 MICRO: International Symposium on Micro-architecture
- 6 HPCA: IEEE Symposium on High-Performance Computer Architecture

## Rank 2

- 1 SUPER: ACM/IEEE Supercomputing Conference
- 2 ICS: International Conference on Supercomputing
- 3 ISSCC: IEEE International Solid-State Circuits Conference
- 4 HCS: Hot Chips Symposium
- 5 VLSI: IEEE Symposium VLSI Circuits

## 系统技术(System Technology)

### Rank 1

- 1 SIGCOMM: ACM Conference on Communications Architectures, Protocols and Applications
- 2 INFOCOM: IEEE International Conference on Computer Communications
- 3 SPAA: Symposium on Parallel Algorithms and Architecture
- 4 PODC: ACM Symposium on Principles of Distributed Computing
- 5 PPOPP: Principles and Practice of Parallel Programming
- 6 MassPar: Symposium on Frontiers of Massively Parallel Processing
- 7 RTSS: Real Time Systems Symposium
- 8 SOSP: ACM SIGOPS Symposium on Operating System Principles
- 9 SOSDI: Usenix Symposium on Operating System Design and Implementation
- 10 CCS: ACM Conference on Computer and Communications Security
- 11 IEEE Symposium on Security and Privacy
- 12 MOBICOM: ACM International Conference on Mobile Computing and Networking
- 13 USENIX Conference on Internet Technologies and Systems
- 14 ICNP: International Conference on Network Protocols
- 15 OPENARCH: IEEE Conference on Open Architectures and Network Programming



16 PACT: International Conference on Parallel Architecture and Compilation Techniques

## Rank 2

- 1 CC: Compiler Construction
- 2 IPDPS: International Parallel and Dist Processing Symposium
- 3 IC3N: International Conference on Computer Communications and Networks
- 4 ICPP: International Conference on Parallel Processing
- 5 ICDCS: IEEE International Conference on Distributed Computing Systems
- 6 SRDS: Symposium on Reliable Distributed Systems
- 7 ASAP: International Conference on Application-Specific Systems, Architectures and Processors
- 8 Euro-Par: European Conference on Parallel Computing
- 9 Fast Software Encryption
- 10 Usenix Security Symposium
- 11 European Symposium on Research in Computer Security
- 12 WCW: Web Caching Workshop
- 13 LCN: IEEE Annual Conference on Local Computer Networks
- 14 IPCCC: IEEE International Phoenix Conference on Computing and Communications
- 15 CCC: Cluster Computing Conference
- 16 ICC: International Conference on Communications

## 编程语言与软件工程(Programming Languages and Software Engineering)

### Rank 1

- 1 POPL: ACM-SIGACT Symposium on Principles of Programming Languages
- 2 PLDI: ACM-SIGPLAN Symposium on Programming Language Design and Impl
- 3 OOPSLA: Object-Oriented Programming Systems, Languages, and Applications
- 4 ICFP: International Conference on Function Programming
- 5 JICSLP/ICLP/ILPS: (Joint) International Conference/Symposium on Logic Programming
- 6 ICSE: International Conference on Software Engineering
- 7 FSE: ACM Conference on the Foundations of Software Engineering (inc: ESEC-FSE when held jointly)
- 8 FM/FME: Formal Methods, World Congress/Europe
- 9 CAV: Computer Aided Verification



## Rank 2

- 1 TACAS: Tools and Algorithms for the Construction and Analysis of Systems
- 2 ESOP: European Conference on Programming
- 3 ICCL: IEEE International Conference on Computer Languages
- 4 PEPM: Symposium on Partial Evaluation and Program Manipulation
- 5 SAS: Static Analysis Symposium
- 6 CP: International Conference on Principles and Practice of Constraint Programming
- 7 RTA: Rewriting Techniques and Applications
- 8 ESEC: European Software Engineering Conference
- 9 IWSSD: International Workshop on Software Specification and Design
- 10 CAiSE: International Conference on Advanced Information System Engineering
- 11 ITC: IEEE International Test Conference
- 12 IWCASE: International Workshop on Computer-Aided Software Engineering
- 13 SSR: ACM SIGSOFT Working Conference on Software Reusability
- 14 SEKE: International Conference on Software Engineering and Knowledge Engineering
- 15 ICSR: IEEE International Conference on Software Reuse
- 16 ASE: Automated Software Engineering Conference
- 17 PADL: Practical Aspects of Declarative Languages
- 18 ISRE: Requirements Engineering
- 19 ICECCS: IEEE International Conference on Engineering of Complex Computer Systems
- 20 IEEE International Conference on Formal Engineering Methods
- 21 International Conference on Integrated Formal Methods
- 22 FOSSACS: Foundations of Software Science and Computation Structures

## 数据库(Data Bases)

### Rank 1

- 1 SIGMOD: ACM SIGMOD Conference on Management of Data
- 2 PODS: ACM SIGMOD Conference on Principles of Database Systems
- 3 VLDB: Very Large Data Bases
- 4 ICDE: International Conference on Data Engineering
- 5 ICDDT: International Conference on Database Theory





## Rank 2

- 1 SSD: International Symposium on Large Spatial Databases
- 2 DEXA: Database and Expert System Applications
- 3 FODO: International Conference on Foundation on Data Organization
- 4 EDBT: Extending Database Technology
- 5 DOOD: Deductive and Object-Oriented Databases
- 6 DASFAA: Database Systems for Advanced Applications
- 7 CIKM: International Conference on Information and Knowledge Management
- 8 SSDBM: International Conference on Scientific and Statistical Database Management
- 9 CoopIS - Conference on Cooperative Information Systems
- 10 ER - International Conference on Conceptual Modeling (ER)

## 数据挖掘(Data Mining)

### Rank 1

- 1 KDD: Knowledge Discovery and Data Mining
- 2 ICDM: IEEE International Conference on Data Mining

### Rank 2

- 1 SDM: SIAM International Conference on Data Mining
- 2 PKDD: European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)

## 算法与理论(Algorithms and Theory)

### Rank 1

- 1 STOC: ACM Symposium on Theory of Computing
- 2 FOCS: IEEE Symposium on Foundations of Computer Science
- 3 COLT: Computational Learning Theory
- 4 LICS: IEEE Symposium on Logic in Computer Science
- 5 SCG: ACM Symposium on Computational Geometry
- 6 SODA: ACM/SIAM Symposium on Discrete Algorithms
- 7 SPAA: ACM Symposium on Parallel Algorithms and Architectures
- 8 PODC: ACM Symposium on Principles of Distributed Computing



- 9 ISSAC: International Symposium on Symbolic and Algebraic Computation
- 10 CRYPTO: Advances in Cryptology
- 11 EUROCRYPT: European Conference on Cryptography

## Rank 2

- 1 CONCUR
- 2 ICALP: International Colloquium on Automata, Languages and Programming
- 3 STACS: Symposium on Theoretical Aspects of Computer Science
- 4 CC: IEEE Symposium on Computational Complexity
- 5 WADS: Workshop on Algorithms and Data Structures
- 6 MFCS: Mathematical Foundations of Computer Science
- 7 SWAT: Scandinavian Workshop on Algorithm Theory
- 8 ESA: European Symposium on Algorithms
- 9 IPCO: MPS Conference on integer programming and combinatorial optimization
- 10 LFCS: Logical Foundations of Computer Science
- 11 ALT: Algorithmic Learning Theory
- 12 EUROCOLT: European Conference on Learning Theory
- 13 WDAG: Workshop on Distributed Algorithms
- 14 ISTCS: Israel Symposium on Theory of Computing and Systems
- 15 ISAAC: International Symposium on Algorithms and Computation
- 16 FST&TCS: Foundations of Software Technology and Theoretical Computer Science
- 17 LATIN: International Symposium on Latin American Theoretical Informatics
- 18 RECOMB: Annual International Conference on Computational Molecular Biology
- 19 CADE: Conference on Automated Deduction
- 20 IEEEIT: IEEE Symposium on Information Theory
- 21 Asiacrypt

## 应用(Applications)

### Rank 1

- 1 I3DG: ACM-SIGGRAPH Interactive 3D Graphics
- 2 SIGGRAPH: ACM SIGGRAPH Conference
- 3 ACM-MM: ACM Multimedia Conference



- 4 DCC: Data Compression Conference
- 5 SIGMETRICS: ACM Conference on Measurement and Modelling of Computer Systems
- 6 SIGIR: ACM SIGIR Conference on Information Retrieval
- 7 PECCS: International Conference on Pervasive and Embedded Computing and Communication Systems
- 8 WWW: World-Wide Web Conference

## Rank 2

- 1 EUROGRAPH: European Graphics Conference
- 2 CGI: Computer Graphics International
- 3 CANIM: Computer Animation
- 4 PG: Pacific Graphics
- 5 IEEE-MM: IEEE International Conference on Multimedia Computing and Systems
- 6 NOSSDAV: Network and Operating System Support for Digital Audio and Video
- 7 PADS: ACM Conference on Principles of Advanced Discrete Simulation
- 8 WSC: Winter Simulation Conference
- 9 ASS: IEEE Annual Simulation Symposium
- 10 MASCOTS: IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems
- 11 PT: Performance Tools - International Conference on Model Techniques and Tools for CPE
- 12 NetStore - Network Storage Symposium

## 人工智能及相关领域(Artificial Intelligence and Related Subjects)

### Rank 1

- 1 AAAI: American Association for Artificial Intelligence National Conference
- 2 CVPR: IEEE Conference on Computer Vision and Pattern Recognition
- 3 IJCAI: International Joint Conference on Artificial Intelligence
- 4 ICCV: International Conference on Computer Vision
- 5 ICML: International Conference on Machine Learning
- 6 KR: International Conference on Principles of Knowledge Representation and Reasoning
- 7 NIPS: Neural Information Processing Systems
- 8 UAI: Conference on Uncertainty in Artificial Intelligence



- 9 ICAA: International Conference on Autonomous Agents
- 10 ACL: Annual Meeting of the ACL (Association of Computational Linguistics)

## Rank 2

- 1 AID: International Conference on Artificial Intelligence in Design
- 2 AI-ED: World Conference on Artificial Intelligence in Education
- 3 CAIP: International Conference on Computer Analysis of Images and Patterns
- 4 CSSAC: Cognitive Science Society Annual Conference
- 5 ECCV: European Conference on Computer Vision
- 6 EAI: European Conference on Artificial Intelligence
- 7 EML: European Conference on Machine Learning
- 8 GP: Genetic Programming Conference
- 9 IAAI: Innovative Applications in Artificial Intelligence
- 10 ICIP: International Conference on Image Processing
- 11 ICNN/IJCNN: International (Joint) Conference on Neural Networks
- 12 ICPR: International Conference on Pattern Recognition
- 13 ICDAR: International Conference on Document Analysis and Recognition
- 14 ICTAI: IEEE conference on Tools with Artificial Intelligence
- 15 AMAI: Artificial Intelligence and Maths
- 16 DAS: International Workshop on Document Analysis Systems
- 17 WACV: IEEE Workshop on Applications of Computer Vision
- 18 COLING: International Conference on Computational Linguistics
- 19 EMNLP: Empirical Methods in Natural Language Processing

## 其他(Miscellaneous)

## Rank 2

- 1 AMIA: American Medical Informatics Annual Fall Symposium
- 2 DNA: Meeting on DNA Based Computers

