代　号	**007**	**学　号**	**1203121619**
分 类 号	**1-1(分类号)**	**密　级**	**公开**

题 (中、英文) 目 基于压缩后缀数组的短读比对算法

A Short Read Aligment Algorithm with Compressed Suffix Array

作 者 姓 名 李双江 **指导教师姓名、职务** 霍红卫

学 科 门 类 工科 **学科、专业** 计算机软件与理论

提交论文日期 二〇一四年十月

Contributers:

HaoChen write the statement page

Olorin183795 bug report

yzg3307 bug report

Author by Xue-Jilong (xuejilong@gmail.com) and Justin-Wong (bigeagle@xdlinux.info)

Typeset by LaTeX 2_ε and CTEX and provide for Bachelor Thesis of Xidian University.

西安电子科技大学

学位论文独创性（或创新性）声明

秉承学校严谨的学分和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关的法律责任。

本人签名：_____ 日期_____

西安电子科技大学

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律署名单位为西安电子科技大学。(保密的论文在解密后遵守此规定)

本学位论文属于保密，在___年解密后适用本授权书。

本人签名：_____ 日期_____

导师签名：_____ 日期_____

摘　要

本文介绍了西电版的 LaTeX 本科毕业设计论文模板，该模板是基于 CTeX 中文宏包开发，指在为西安电子科技大学的本科毕业生提供一个简单、专业、有效的排版工具，且该版本不打算加入研究生毕业论文和博士生毕业论文，因为定制模板也是一个很复杂的事情，如果有可能的话，后期可能继续单独写研究生和博士生的 LaTeX 模板。作者本着为西电同学服务的原则开发，并不承担一切有关责任与义务，如维护、更新等，但欢迎提交 BUG 。祝西安电子科技大学的同学前程似锦。

西安电子科技大学是以信息与电子学科为主，工、理、管、文多学科协调发展的全国重点大学，直属教育部，是国家“211 工程”立项建设的重点高校之一，是全国 56 所设有研究生院的高校之一，37 所示范性软件学院的高校之一，也是全国 20 所获批设立集成电路人才培养基地的高校之一。

1931 年诞生于江西瑞金的中央军委无线电学校，是毛泽东等老一辈革命家亲手创建的第一所工程技术学校。1958 年学校迁址西安，1966 年转为地方建制，1988 年定为现名。建校 79 年来，学校始终得到了党和国家的高度重视，是我国“一五”重点建设的项目之一，也是 1959 年中央首批批准的全国 20 所重点大学之一。20 世纪 60 年代，学校就以“西军电”之称蜚声海内外。毛泽东同志曾先后两次为学校题词：“全心全意为人民服务”、“艰苦朴素”。学校现建设有南北两个校区，总占地面积 4000 亩，校舍建筑面积 130 多万平方米，图书馆藏书近 420 万册[7]。

校现有各类在校生 3 万余人，其中博士研究生 1700 余人，硕士研究生 8100 余人，设有通信工程学院、电子工程学院、计算机学院、机电工程学院、技术物理学院、经济管理学院、理学院、人文学院、示范性软件学院、微电子学院、国际教育学院、生命科学技术学院、网络与继续教育学院以及长安学院等 14 个学院。

关键词： 西电，论文，毕业设计，模板

ABSTRACT

This page is English abstract test.English is a West Germanic language that arose in the Anglo-Saxon kingdoms of England and spread into what was to become south-east Scotland under the influence of the Anglian medieval kingdom of Northumbria. Following the economic, political, military, scientific, cultural, and colonial influence of Great Britain and the United Kingdom from the 18th century, via the British Empire, and of the United States since the mid-20th century, it has been widely dispersed around the world, become the leading language of international discourse, and has acquired use as lingua franca in many regions. It is widely learned as a second language and used as an official language of the European Union and many Commonwealth countries.

This page is English abstract test.English is a West Germanic language that arose in the Anglo-Saxon kingdoms of England and spread into what was to become south-east Scotland under the influence of the Anglian medieval kingdom of Northumbria. Following the economic, political, military, scientific, cultural, and colonial influence of Great Britain and the United Kingdom from the 18th century, via the British Empire, and of the United States since the mid-20th century, it has been widely dispersed around the world, become the leading language of international discourse, and has acquired use as lingua franca in many regions. It is widely learned as a second language and used as an official language of the European Union and many Commonwealth countries.

This page is English abstract test.English is a West Germanic language that arose in the Anglo-Saxon kingdoms of England and spread into what was to become south-east Scotland under the influence of the Anglian medieval kingdom of Northumbria.

Keywords: Xidian, University, Thesis, Template

目　录

第一章　绪论

1.1　研究意义和背景介绍

DNA(脱氧核糖核酸) 是生物遗传信息的载体，其双螺旋结构的两个链互相补充，构成稳定结构。其中每个链都含有完备的遗传信息，这些遗传信息体现在构成 DNA 链的四种碱基——腺嘌呤（A），胸腺嘧啶（T），鸟嘌呤（C）和胞嘧啶 (G) 的排列顺序上。在现代生物学研究中，为分析 DNA 的遗传表达等特性，需要特定对物种 DNA 进行测序。早期的 sanger 测序作为第一代测序手段在人类基因组计划中起到了巨大的作用。

随着生物学，医学等相关科学的发展，新的 DNA 测序技术不断涌现，其中，以 Illumina/Solexa 为代表的 NGS(NEXT-GENERATION SEQUENCING DAT) 技术以其低廉的测序成本和便捷快速的特点成为当前的主流 DNA 测序技术。基于这一新技术实现的测序机器每台工作一天就能产生数十亿的短读序列 (short reads)[15]。NGS 测序技术一般应用于两类测试场景，重测序 (Resequencing) 和从头测序 (de novo sequencing)，这也对应着产生了 DNA 分析领域的两个最核心的研究问题：比对 (alignment) 和重组 (assembly)。若测序的目标物种的基因序列之前还从未被测序过，那么从头测序就是研究的第一步，这需要关注把短读以最优方式连接起来。若测序目标物种已经完成了测序，那么重测序关注的问题是如何把短读序列映射到已知的同物种基因组上，从而分析同源生物的个体基因差异，这个过程就是本文关注短读比对 (short read alignment)。由于每一次测序实验都会得到大量的短读 (short reads) 序列 (5 亿到 20 亿个)，同时生物个体基因之间的差异会导致基因序列存在差异，短读映射面临着基因的近似比对和快速高效比对两个难题。本文即提出一种基于压缩后缀数组索引算法的快速高效比对算法来解决这两个问题。

重测序得到的短读序列中每一个短读一般不超过 1000 个碱基 (大多数情况下都是 20 到 100 个碱基的长度), 但一次测序实验中短读数量都会超过一千万个。参考序列是已经经过准确测序，重组后的已知基因组序列，比如人类基因组序列就是合并出来的总长达 2.8G 的 DNA 序列。出于医疗，身份鉴别等原因会对某个具体的人进行再次 DNA 测序，这就是 DNA 重测序，此时测序得到的大量短读序列分析的第一步就是把这些短读映射到参考序列上，对人类而言，大多都是映射到人类基因组序列上，也可以映射到一个人工合成的参考序列上。映射的过程是对每一个短读在参考序列上查找的过程，即要在参考序列上找到一个合适的位置，使得从这个位置开始，短读是参考序列的一个子串。

综上所述，短读序列的比对问题可以抽象为一个模式查找问题：给定一个共有 m 个模式的模式集合 $P = \{P_1, P_2 \dots P_m\}$，每个模式的长度已知分别为 $l_1, l_2 \dots l_m$，已知一个长为 n 的参考序列 T，求得一个集合 $S = \{s_1, s_2 \dots s_n\}$ 使得

$P_i = T[s_i \dots s_i + l_i - 1]$。这个查找的过程即为短读到参考序列的比对映射。其中参考序列 T 和短读序列 P_i 都是由 DNA 测序中常用的碱基字符 $\{A, T, C, G, N\}$ 构成的。

1.2 国内外研究现状

为实现快速且准确的短读序列映射，近年来出现了很多比对算法。所有这些算法都可以分为两类，一类是通过对短读序列使用散列表等方法建立短读序列的索引，之后遍历整个参考序列。另一类是为参考序列建立索引，之后再对每个短读进行独立的比对。

第一类比对算法的代表是 MAQ，ZOOM，SHRiMP 等。MAQ[8] 基于散列技术，结合短读中每一个核苷酸的测序质量分数，实现了无空位 (ungapped) 比对。ZOOM[12] 使用了 space seeds 技术，提高了比对的精确率。而 SHRiMP[17] 则结合 space seeds" 和 smith-water 算法得到了更高的精确率。

第二类算法为参考序列建立索引，通过索引后的数据可以实现快速的比对。如 SOAP,WHAM,BFAST 等。SOAP[9] 使用 seeds 技术和一个散列查询表加速比对，且可以处理较少的空位比对。WHAM[11] 对参考序列建立散列表，先通过散列表查找潜在的比对位置，再进一步比对确定最终结果。BFAST 则通过为参考序列建立多个索引来提高精确度。这几种方法使用的索引方法都需要很大的内存空间，所以比对时空间需求很大，尤其是在用类基因组这样的较大序列作为参考序列时。在第二类方法中以 SOAP2,Bowtie,BWA 为代表的基于 BW 变换 (Burrows-Wheeler transform,BWT)[1] 来创建参考序列索引的方法具有很大的空间优势。Bowtie[5] 使用 BWT 建立索引，采用回溯递归的搜索方法，再结合双端搜索实现了高速，空间高效的比对，是目前最快的比对软件之一，但缺陷是不能实现空位 (gap) 比对。BWA[6] 也是基于 BWT 的一种比对算法，比对速度较 Bowtie 慢，但可实现空位比对。SOAP2[10] 使用了 bidirectional BWT 来建立参考序列的索引，比对速度和 Bowtie 相当。基于 BWT 的这些方法都使用了后向搜索方法[13] 来加速查询。后向搜索可以在 $O(m)$ 时间内实现长为 m 的字符串的计数查询，以及 $O(m \log n)$ 时间复杂度的 query 查询。利用后向搜索的性质，Bowtie 实现了基于回溯法的非精确匹配算法，而 BWA 则采用前缀树搜索的方法实现非精确匹配。在实现非精确匹配的基础上，加上一些打分机制，既实现了短读序列到参考序列的匹配。

1.3 本文的主要内容及组织结构

本文提出一种采用压缩后缀数组 (COmpressed Suffix Array,CSA) 建立索引[2]，实现短读比对的算法:CSAA(csa alienment)。这一算法采用的是 CSA 的后向搜索特性，同时还使用了优先队列来保存所有可能的匹配位置，并为每个可能的匹配位置打分，在匹配过程中，通过分支限界抛弃所有低分搜索方向，降低搜索空间，

同时保证匹配结果最优。按照上一节中对短读比对算法的分类，该算法属于对参考序列进行索引的比对算法。

第二章　预备知识

以 BWT 为代表的自索引算法近年来在序列比对领域多有出现，例如前文中提到的 Bowtie，BWA 等。本文提出的序列比对算法使用的也是一种自索引算法：压缩后缀数组 (CSA)[2]。同 BWT 相比，CSA 的模式查询速度更快，应用于序列比对，相应的比对速度也会更快，提高整个比对的效率。相应的，在使用 CSA 之前有必要对 CSA 的一些概念进行一些简单的叙述，此外序列比对领域的一些基本概念也会在本章中解释。

2.1　压缩后缀数组和模式匹配

2.1.1　后缀数组和压缩后缀数组简介

压缩后缀数组 (CSA) 是由 Grossi 和 Vitter[2] 最早提出的第一种实现全文索引的压缩索引数据结构，是对后缀数组 (SA)[14] 占用空间过大的改进，并且实现了自索引特性。

设长为 n 的文本序列 T，字符集为 Σ，本文中将假设 T 有一个特殊的结尾符号 \$，\$ 不在 Σ 中并且字典序小于 Σ 中的所有符号。假设 T 存储在一个数组 $T[0\ldots n-1]$ 中。对任何的整数 i，假设

- $T[i]$ 为 T 中从左往右 0 开始的第 i 个字符;
- T_i 为 T 的第 i 个后缀，即 $T_i = T[i]T[i+1]\ldots T[n-1]$。

的后缀数组 $SA[0\ldots n-1]$ 定义为 T 的 n 个后缀按字典序排序后的序列，由 $\{0,1,\ldots,n-1\}$ 的一个排列构成，满足 $T_{SA[0]} < T_{SA[1]} < \ldots < T_{SA[n-1]}$。即 $SA[i]$ 表示 T 的 n 个后缀中第 i 小的后缀的开始位置。如表2.1所示。后缀数组占用空间 $n\log n$，给定文本 T 和其后缀数组 $SA[0\ldots n-1]$，T 中的任何模式 P 可以在 $O(|p|\log n + occ)$ 时间复杂度内求出其出现位置[14]，并且不需要读原文本 T。其中 occ 是模式的出现次数。

对于任意的整数 $i \in [0\ldots n-1]$，定义 $SA^{-1}[i] = j$ 使得 $SA[j] = i$，很明显 $SA^{-1}[i]$ 为 T_i 在 T 的所有后缀中的排名，即 T 的后缀中比 T_i 小的后缀的数量。

$$\Phi[i] = j \qquad if\ SA[j] = (SA[i]+1) \mod n \text{[3]} \tag{2-1}$$

序列 T 的压缩后缀数组 (Compressed Suffix Array, CSA) 是对后缀数组 (SA) 空间复杂度过大的一个改进。其本身也是一个包含 n 个整数与后缀数组 SA 大小相同且由 SA 的近邻函数变换而来的数组 Φ。近邻函数定义如2-1，由于 $T[n-1] = \$$，所

表 2.1 *acaaccg*\$ 的后缀数组和 Φ 数组

i	$T[i]$	T_i	$SA[i]$	$T_{SA[i]}$	$\Phi[i]$	$T[SA[i]]$
0	a	acaaccg\$	7	\$	2	\$
1	c	caaccg\$	2	aaccg\$	3	a
2	a	aaccg\$	0	acaaccg\$	4	a
3	a	accg\$	3	accg\$	5	a
4	c	ccg\$	1	caaccg\$	1	c
5	c	cg\$	4	ccg\$	6	c
6	g	g\$	5	cg\$	7	c
7	a	\$	6	g\$	0	g

以 $\Phi[0] = SA^{-1}[0]$。另一个角度来看，若后缀 T_k 在 T 的后缀中排名为 i，则 $\Phi[i]$ 为后缀 T_{k+1} 在 T 的后缀中的排名。同时，可以看到 $SA^{-1}[1] = SA^{-1}[SA[SA^{-1}[0]+1] = \Phi[\Phi[SA[SA^{-1}[0]]]] = \Phi[\Phi[0]]$，同理可以得到 $SA^{-1}[2] = \Phi[\Phi[\Phi[0]]]$。以此类推，即可根据 $\Phi[0\ldots n-1]$ 迭代求出 $SA^{-1}[0\ldots n-1]$，由 $SA^{-1}[0\ldots n-1]$ 可快速求出 $SA[0\ldots n-1]$。由此可得出，从后缀数组 $SA[0\ldots n-1]$ 到数组 $\Phi[0\ldots n-1]$ 的变换是可逆的。

$\Phi[0\ldots n-1]$ 包含 n 个整数，显示存储时，也需要 $n\lceil\log n\rceil$ 位的存储空间，同后缀数组 SA 相同。然而，观察表2.1 可以发现 $\Sigma[1\ldots n-1]$ 可以分解为 $|\Sigma|$ 个严格递增的序列，这使得压缩后缀数组可以用简明数据结构存储。而 $\Sigma[1\ldots n-1]$ 的递增属性则是基于以下引理。

引理 2.1. *对于任意的整数 $i<j$, 若 $T[SA[i]] = T[SA[j]]$，则 $\Phi[i] < \Phi[j]$。*

证明. 当 $i<j$ 时，则 $T_{SA[i]} < T_{SA[j]}$ 一定成立，反之亦然。这等价于当 $T[SA[i]] = T[SA[j]]$ 时，$T_{SA[i]+1} < T_{SA[j]+1}$，即 $T_{SA[\phi[i]} < T_{SA[\Phi[j]]}$，所以可以得到 $\Phi[i] < \Phi[j]$。即引理2.1成立。 □

对任意一个 Σ 中的字符 c，定义 $\alpha(c)$ 为 T 的后缀中首字符小于 c 的后缀的数目，定义 $\beta(c)$ 为 T 的后缀中首字符为 c 的后缀的数目。则有以下结论：

推论 2.2. *对于 Σ 中的任意一个字符 c，$\Phi[\alpha(c)], \Phi[\alpha(c)+1]\ldots\Phi[\alpha(c)+\beta(c)-1]$ 是一个严格递增序列。*

证明. 对于任意的字符 c，$T[SA[\alpha(c)]] = T[SA[\alpha(c)+1]] = \ldots = T[SA[\alpha(c)+\beta(c)-1]] = c$，由引理2.1可知，$\Phi$ 在 $\Phi[\alpha(c)\ldots\alpha(c)+\beta(c)-1]$ 上严格递增。 □

根据以上结论，Φ 可以划分为 $|\Sigma|$ 个递增序列，Grossi 和 Vitter[2] 提出了一种压缩模式来存储 Φ，使得可以在 $O(n(H_0+1))$ 位的空间内存储 Φ 数组，其中 $H_0 \leq \log|\Sigma|$，是文本 T 的 0 阶经验熵。这种存储模式就是下文中叙述的简明数据结构。

2.1.2　简明数据结构

简明数据结构 (Succint Data Structure) 是对整数序列进行简明编码，达到压缩存储的效果并实现常数时间解码的数据结构。本节中将以 Vitter 原始论文中的 Rice 编码为例阐述简明数据结构的存储原理。实际上，除了 Rice 编码，简明数据结构还可以使用很多编码形式，如 $\Delta-\sigma$ 编码等。

设有 s 个升序的整数，每一个整数有 w 位，$s<2^w$。简明数据结构的原理是把这 s 个整数分为两部分，分别存储在两个表 Q,R 里。取出每个整数的前 $z=\lfloor\log s\rfloor$ 位组成一个新的整数，设为 q_i，明显有 $0\leq q_h\leq q_{h+1}<s$，其中 $1\leq h<s$。设各个整数中去除前 z 位后剩下的部分组成的整数为 $r_1,r_2,\cdots r_s$。

由于 $q_1\leq q_2\leq\cdots\leq q_s$，所以采用一元编码（unary repesentation）表示 q_i。对于任意的整数 $i\geq 0$，其一元编码为 0^i1，即 i 个 0 后紧随一个 1。在此构建 Q 表采用一元编码表示：$q_1,q_2-q_1,\cdots,q_s-q_{s-1}$。由此，表 Q 是一个二进制表。加上辅助数据结构 $select$ 操作，可以在常数时间内获得表 Q 二进制串中第 h 个 1 出现的位置。为获取 q_h，只需调用 $select(h)$ 获得第 h 个 1 出现的位置 j，再通过 $j-h$ 计算出二进制串中前 j 位中 0 的个数，很明显，串中 0 的个数 $j-h$ 即为 q_h。

表 Q 由两部分组成，表示 q_i 的二进制串和辅助数据结构。总共有 s 个数，所以二进制串中至少有 s 个 1；q_i 中最大的数为 2^z，所以最多有 2^z 个 0，所以二进制串的内存空间为 $s+2^z\leq 2s$ 位。而支持 $select$ 操作的辅助数据结构的空间复杂度是 $O(s/\log\log s)$ 位，所以 Q 表总的空间复杂度是 $2s+O(s/\log\log n)$。查询时间为常数时间。

对于 R 表，可以简单的当作普通数组存储即可，总共需要 $s(w-\lfloor\log s\rfloor)$ 位，查询时间也为常数时间。

最后，为查询升序整数序列中任意一个整数 s_h，只需查询 Q 表和 R 表分别获取 q_h 和 r_h，而后返回 $q_h\cdot 2^{w-z}+r_h$ 即为所查询的 s_h。时间复杂度为常数时间。

综上所述，可得以下结论：

推论 2.3. *对于 s 个升序的整数组成的序列，设每一个整数最多 w 位且 $s<2^w$，可以把这 s 个整数存储在最多 $s(2+w-\lfloor\log s\rfloor)+O(s/\log\log s)$ 位的空间内，且查询任意整数的时间复杂度为 $O(1)$。*

2.1.3 rank&select 操作

根据上一小节的论述，简明数据结构实现的基础是 rank&select 操作，本节即详细介绍这两个操作的实现方法。Jacobson 在论文[4] 中阐述了这两种操作的经典采样分割实现方法。由于经典方法存在空间复杂度较低的问题，本文采用了更高效的 RRR 方法实现 rank&select 操作。

RRR 方式是由 R.Raman，V.Ramna 以及 S.Srinivasa Rao 等人于 2002 年提出的一种静态的字典结构[16]。通过这种结构可以实现对 01 二元序列的常数时间的 rank&select 操作，并且采用同一方法可扩展到对多符号序列的常数时间的 rank&select 操作。在二元 01 序列上，RRR 方法实现 rank 操作只需要 $nH_0 + o(n)$ 位的空间，而常用的 Jacobson 的 rank&select 方法则需要 $n + O(n \log\log n/\log n)$ 位的空间[4]。可以看到在 01 序列中，如果 0 和 1 的几率相等时，二者的空间占用相差不大，而若 0 和 1 出现的几率并不相等，其中一个出现的几率远大于另一个时，RRR 方法的空间占用将小于 n 比特。而在压缩后缀数组 $\Phi[0 \ldots n-1]$ 的简明存储中，需要维持一个由 01 序列构成的采样点的字典结构，该字典需要实现 rank&select 操作，且字典中的 0 远多于 1，在这一应用场景中，采用 RRR 方法要优于 Jacobson 的方法。

RRR 方法和 Jacobson 的方法在目录结构上类似，都采用了分块的方法和两层目录结构。具体做法是首先把长为 n 的二元串 B 分长为 $s = \log n^2$ 的大块 $S_1, S_2 \ldots S_{n/s}$。之后每一个大块再分成长为 $b = \log n/2$ 的小块 $B_i(j)$。这两种划分方法在 RRR 中和 Jacobson 的方法是一致的，所不同的是之后的处理。对于每一个小块 $B_i(j)$，在 Jacobson 的方法中是显式直接存储的，而在 RRR 方法中则采用了一个 (c, o) 对替代 $B_i(j)$，其中 c 表示 $B_i(j)$ 这个小块中的 1 的个数，即这个小块的类别，而 o 表示 $B_i(j)$ 在所有的有 c 个 1 的长为 b 位的证数中的名次。显然，对于第 c 类，总共有 $\binom{b}{c}$ 个。而 c 的最大值为 b，所以一个 (c, o) 对需要的存储空间是 $\log c + 1 + \log \binom{c}{b}$ 位。从这里可以看出，相对于原来的 b 位的原始串，采用一个 (c, o) 对替代后，所需空间是随原始串中 1 的个数变化的，1 的个数越少（即 c 越小），所需要的存储空间也会相应的变小，而就整体而言，一个 (c, o) 对所占的空间也是小于 b 位的。这就是 RRR 方法的优势所在。对于每一类 c 中的每一个 (c, o) 对，都可以预先处理得到这个 (c, o) 对对应的长为 b 的 01 串的每一位的 $rank$ 值，并保存为 G_c 表，总共需要 $b\log(c+1)$ 位的空间。所有的 G_c 表组合起来就构成了我们预处理得到的表 G，总共需要 $\sum_{c=0}^{b} b\log(c+1)\binom{c}{b} = O(\sqrt{n}poly\log n)$ 位的空间。

通过上面叙述的方法，可以把每一个小块 $B_i(j)$ 变换成一个 (c, o) 对，表示为 $D_i(j)$，并且 $D_i(j)$ 总共需要 $\log(c+1) + \log \binom{c}{b}$ 位的空间，累加所有的小块对应的 (c, o) 对所需要的空间，前面一项累加后为 $O(\log\log n)$ 位，后面一项累加起来为 nH_0 位。把所有的变长的 $D_i(j)$ 连接起来构成一个单独的表，即 D 表，所需空间为 $nH_0 + O(\log\log n)$ 位。

对于每一个大块 S_i，对应存储一个指针 P_i 指向这个大块的第一个小块对应的 (c,o) 对在 D 表中的位置，即 $P_i = D_i(0)$，并且指针 R_i 存储这个大块对应的第一位的 $rank$ 值，即 $R_i = rank((i-1)*s)$。P 表和 R 表总共需要的空间是 $O(n/\log n)$ 位。同样的，对应每一个属于大块 S_i 的小块 $B_i(j)$，也存储一个指向其 (c,o) 对在 D 表中的位置的指针 $L_i(j)$，只是 $L_i(j)$ 是该位置相对于所在大块位置的相对位置，即 $L_i(j) = D_i(j) - P_i$。类似的，保存每一个小块 $B_i(j)$ 的第一位的 $rank$ 值 $Q_i(j)$，当然也是相对于所在大块的 $rank$ 值，即 $Q_i(j) = rank((i-1)*s)+(j-1)*b) - R_i$。由于这些相对量的最大值都是 $\log n$，所以 L 表和 Q 表总共需要 $O(n\log\log n/\log n)$ 位的空间。

在求解任意的 $rank(p)$ 时，首先计算第 p 位对应的大块的编号 $i = p/s$，以及小块编号 $j = (p-(i-1)*s)/b$，之后，加上所在的大块对应的 $rank$ 值 R_i 和小块对应的相对 $rank$ 值 $Q_i(j)$。再根据 P_i 和 $L_i(j)$ 的值可得到这个小块对应的 (c,o) 对的值 $D_i(j)$，通过访问 D 表的 $D_i(j)$ 位置，即可得到第 p 位所在小块的每一位的 $rank$ 值，加上前面得到的相对 $rank$ 值即可得到最终的 $rank$ 值。

上面所述即为 **RRR** 方法的原理，总共需要保存 D,P,R,L,Q 五个表，总的空间需求是 $nH_0 + O(n\log\log n/\log n)$ 位，可以在常数时间内实现 $rank$ 操作。

RRR 方法的 $select$ 操作的实现是基于 $rank$ 的实现的，查找 $rank[j] = i$，并且 $rank[j-1] = i-1$，则有 $select[i] = j$。所以，只需在一直 $rank$ 时，进行简单的二分查找即可实现求解 $select$ 操作，且不需要任何的额外的辅助空间，时间复杂度是 $rank$ 操作时间复杂度的 $\log n$ 倍。该方法的优点是实现简单，不需要额外的空间，但却并没有很好的利用 **RRR** 方法的性质。从上一小节的叙述中可知，为了实现 **RRR** 的 $rank$ 操作，特意存储了两个目录表，R 表和 Q 表，其中 R 表是第一级目录，即大块儿的初始位置的 $rank$ 值，而 Q 表是第二级目录，即各个小块儿的初始位置相对所在大块儿的起始位置的相对 $rank$ 值。利用这一性质，$select$ 操作可以更高效的完成，原理依然是二分搜索，但无需对整个序列的 $rank$ 进行二分操作，而是在两层目录上分别进行二分搜索，逐层的缩小搜索的范围，最后实现 $select$ 操作。具体的计算 $select[j]$ 的算法过程如下：

算法 2.1.

1. 首先搜索 R 表，得到一个位置 i，使得 $R[i] < j < R[i+1]$，即可确定 $i*s < select[j] < (i+1)*s$。

2. 再搜索 Q 表中的 $Q[i*s, i*s+1 \ldots (i+1)*s]$ 得到 k 使得 $R[i]+Q[k] < j < R[i]+Q[k+1]$，即可确定 $k*b < j < (k+1)*b$。

3. 有了前两部的范围，实际上即得到了对应的第三次查询的需要的大块儿的编号 i 和小块儿的编号 k，查询 $P[i]$ 和 $L[j]$ 即得到了对应的 (c,o) 对在 D 表中的位置，接下来查询 D 表即可得到该 (c,o) 对对应的局部 $rank$ 值。

4. 线性查询第 3 步中得到的 D 表中的 $rank$ 序列，使得 $rank[m] = j - R[i] - Q[k]$，则 $select[j] = i * s + k * b + m$，即为最终查询结果。

上述的 $select$ 方法基于二分实现，时间复杂度是三次查询时间复杂度之和，即 $\log n/s + \log s/b + \Theta(b)$。

2.2 序列比对

2.2.1 DNA 序列格式

在 DNA 序列分析领域，DNA 数据一般都来自国际知名的几大 DNA 数据库，如 GenBanki，EMBL，DDBJ 等。不同的测序方法，通常得到的序列数据也会有一些差异。对此，为方便后续处理，生物信息学定义了一些通用的序列存储格式，如 Fasta，Fastq 等。通常 Illumina 测序数据都是 Fastq 格式，所以本文中实现的软件 CSAA 也以 Fastq 作为标准输入格式。

Fastq 格式是 DNA 序列格式中常见的一种，Fastq 格式的序列一般都包含有四行，第一行由'@' 开始，后面跟着序列的描述信息，这点跟 Fasta 格式是一样的。第二行是序列的字符表示。第三行由'+' 开始，后面也可以跟着序列的描述信息，和第一行信息相同，通常可以省略。第四行是第二行序列的质量评价（quality values，是测序的质量评价），字符数跟第二行的序列是相等的。下面是 Fastq 格式序列的一个序列示例。

```
@HWUSI-EAS100R:6:73:941:1973#0/1
GATTTGGGGTTCAAAGCAGTATRRRGYKKKMSTCAAATAGTAAATCCATTTGTTCAACT
+HWUSI-EAS100R:6:73:941:1973#0/1
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC6
```

Illumina 测序仪是按照荧光信号来判断所测序的碱基是哪一种的，例如红黄蓝绿分别对应 ATCG，但对每个结果都是有一定的误差的。最初 sanger 中心用 Phred quality score 来衡量该 read 中每个碱基的质量，既 $Q = -10\lg P$，其中 P 代表该碱基被测序错误的概率，如果该碱基测序出错的概率为 0.001，则 $Q = 30$，30+33=63，63 对应的 ASCii 码为“?”，则在第四行中该碱基对应的质量分数代表值即为“?”。一般地，碱基质量从 0-40，既 ASCii 码为从“!”（0+33）到“I”(40+33)。这上是 sanger 中心采用记录 read 测序质量的方法，Illumina 没有完全依照 sanger 中心的方法来定义测序质量，而是把 P 换成了 $P/(1-P)$，其他完全按照 sanger 的定义来做。可以看出当测序质量很高的情况下两种形式几乎没区别，但低质量的碱基则有区别了。

在 Fastq 格式中还可能出现其他一些核苷酸符号，具体含义如表2.2所述。

DNA 序列的标准保存格式是 Fastq 等格式，同样的，对序列比对的输出格式，也有一个约定的标准数据格式：SAM 格式。SAM 的全称是 sequence alignment/map

表 2.2 Fastq 格式支持的核苷酸符号

核苷酸代码	意义
A	Adenosine
C	Cytosine
G	Guanine
T	Thymidine
U	Uracil
R	G A (puRine)
Y	T C (pYrimidine)
K	G T (Ketone)
M	A C (aMino group)
S	G C (Strong interaction)
W	A T (Weak interaction)
B	G T C (not A) (B comes after A)
D	G A T (not C) (D comes after C)
H	A C T (not G) (H comes after G)
V	G C A (not T, not U) (V comes after U)
N	A G C T (aNy)
X	masked
-	gap of indeterminate length

format，一般是文本形式的，也可以存为二进制形式文件，即 BAM 格式。SAM 由头文件和 map 结果组成，头文件由一行行以“@”起始的注释构成。而 map 结果是类似下面的文本:

```
C12FP66670 0    chr1  12805 1 42M4I5M * 0 0 TTGGATGCCCCTC...
C12FP30032 272  chr1  13494 1 51M     * 0 0 ACTGCCTGGCGCT...
```

SAM 文件中每个 read 只占一行，被 tab 分成了很多列，一共有 12 列，分别记录了：read 名称，SAM 标记，chromosome 名称，5端起始位置，MAPQ（mapping quality，描述比对的质量，数字越大，特异性越高)，CIGAR 字串 (记录插入，删除，错配信息)，mate 名称 (记录 mate pair 信息)，mate 的位置，模板的长度，read 序列，read 质量，程序用标记。

本文中重点关注的是第三列，chrome 名称，以及第四列，起始位置。通常作为参考序列的基因组是由多条染色体构成的，比对程序需要得到 read 在哪一条染色体上，以及在改染色体上的位置，即 5’ 端起始位置。

2.2.2 单端测序和双端测序

目前的测序方法中，如 Solid，都有单端测序 (Single-read) 和双端测序 (Paired end) 之分。二者再测序方法上不同，得到的测序数据也有一些差异。主要区别在于文库的建立上。

无论是单端测序还是双端测序，第一步都是对 DNA 分子进行切割，这是通过切割酶来实现的。切割后，大 DNA 分子被切割成长为 300bp 左右的短序列 (fragments)。测序第二步是增值，通过对这些短序列进行复制，增值，提高 DNA 分子数量。第三部是加入引物，开始测序。单端测序时只在 DNA 短序列分子的一端加上引物，然后依次读取核苷酸，直到读完一个 read。通常一个 read 长为 80 到 1000bp，读取核苷酸时，因为越往后读取错误率越高，所以一般 read 序列也是越往后，可靠性越低。双端测序时，会在 DNA 短序列两端都加上引物，然后分别读取核苷酸。所以，双端测序得到的是一个 DNA 短序列分子的两个 read，这两个 read 读取的是 DNA 链的两个不同的链，并且因为只读取两端的前 100bp 左右的核苷酸，所以，这两个 read 序列并不一定重合，二者之间有一定的距离 (distance)，distance 的长度为短序列 (fragment) 的长度减去两个短读序列的长度之和。反映到在参考 DNA 序列上，distance 为两个序列映射位置之差的绝对值。

2.3 本章小结

本章分为两个部分，对本文用到的一些先验知识做了一些简述。第一部分简述了本文要用到的索引算法：压缩后缀数组。描述了其基本特性，以及可压缩性，接着对简明数据结构做了一些简单介绍，重点是使用到的 rank&select 结构：RRR 结构。第二部分是对生物信息学领域序列比对的一些基本概念的解释。包括 DNA 序列数据格式和单端测序，双端测序的概念。

第三章　表格图形

3.1　表格

与 word 不同，LaTeX 通过一定的语法规则将表格写成纯文本形式。基本规则包括：表格从上到下，每一行从左到右，单元格内容使用 & 分隔，用 \\ 换行。最基本的表格环境是 tabular 环境。下面是一个简单的表格代码和实际效果：

姓名	年龄
张三	32
李四	12
王五	24

学术论文普遍使用三线表。三线表的特点主要是：整个表格通常只有三条横线，首尾两条横线较粗，中间一条较细，一般不使用竖线。LaTeX 处理三线表相当简单方便。用到的宏包主要是 booktabs 。下面是普通三线表的代码和效果：

表 3.1 示例表格

姓名	年龄	地址
张三	32	中华人民共和国
李四	12	中华人民共和国
王五	24	中华人民共和国

有时三线表需要固定某列的列宽，或者指定整个表格的总宽度，指定某几列自动伸缩。使用 tabularx 宏包可以实现自动伸缩列宽。下面是一个简单的例子。与普通的 tabular 环境不同之处在于：（1）需要指定整个表格的总宽度；（2）需要用 X 指定至少一列为自动伸缩列。见表3.2。

好了，表格的介绍就到此为止，关于表格的学习还有很大的学问，可以找专门的教程去学校，这里只是一个介绍。

3.2　图形

LaTeX 中一般只直接支持插入 eps(Encapsulated PostScript) 格式的图形文件, 因此在图片插入 latex 文档之前应先设法得到图片的 eps 格式的文件. 在 LaTeX 文档中插入图片都是通过使用一些 latex 图形处理宏命令来实现的, 有很多宏命令都支持在在 LaTeX 文档中插入 eps 格式的图形文件。

表 3.2 2000 和 2004 年中国制造业产品的出口份额

	2000	2004
钢铁	3.1	5.2
化学制品	2.1	2.7
办公设备及电信设备	4.5	15.2
汽车产品	0.3	0.7
纺织品	10.4	17.2
服装	18.3	24

3.2.1 图形位于页面中

命令其中的" 高度" 和" 宽度" 是指希望图片打印的高度和宽度, 必须给出单位, 可用厘米 (cm) 或英寸 (in). 高度和宽度也可用上述格式同时给出, 这样可以改变原图的长宽比例. 上述命令中的图片文件名是指欲插入的图片文件的文件名, 图片必需是 eps 格式的. 用 graphicx 包的 includegraphics 宏命令插入图片时还可以使图片旋转。

该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。好了，开始。该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。好了，开始。

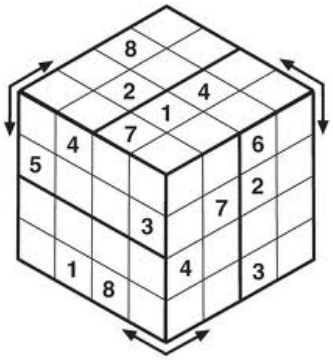

图 3.1 这是一个图片测试例子（中）

该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。好了，结束。

3.2.2 图形位于页面上

该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。好了，结束。该

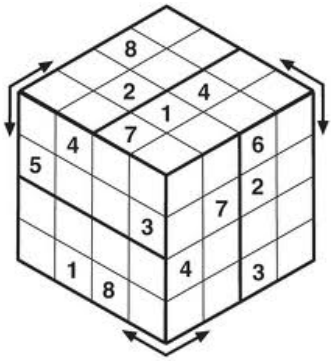

图 3.2 这是一个图片测试例子（上）

页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。好了，结束。

该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。

3.2.3　图形位于页面下

该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。

该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。

该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时

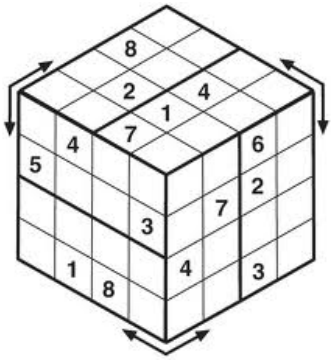

图 3.3 这是一个图片测试例子（下）

间来研究学习下，就像表格一样，下面这只是一个简单的例子。该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。

该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。

该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。

该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。该页是专门用来测试插入图片的方法，这个方法有很多种，需要自己花点时间来研究学习下，就像表格一样，下面这只是一个简单的例子。

第四章　总结与展望

4.0.4　总结

4.0.5　进一步工作

附录 A　西安电子科技大学本科生毕业设计论文撰写规范

A.1　毕业设计（论文）的总体要求

撰写论文应简明扼要，一般不少于 15000 字（外语专业可适当减少，但不得少于 10000 单词，且须全部用外语书写)。

A.2　毕业设计（论文）的编写格式

每一章、节的格式和版面要求整齐划一、层次清楚。其中:

- 论文用纸：统一用 A4 纸，与论文封皮，任务书，工作计划，成绩考核表一致。
- 章的标题：如：‘‘摘要’’、‘‘目录’’、‘‘第一章’’、‘‘附录’’ 等，黑体，三号，居中排列。
- 节的标题：如：“2.1 认证方案’’、“9.5 小结’’ 等，宋体，四号，居中排列。
- 正文：中文为宋体，英文为 “Times News Roman’’，小四号。正文中的图名和表名，宋体，五号。
- 页眉：宋体五号，居中排列。左面页眉为论文题目，右面页眉为章次和章标题。页眉底划线的宽度为 0.75 磅。
- 页码：宋体小五号，排在页眉行的最外侧，不加任何修饰。

A.3　毕业设计（论文）的前置部分

毕业设计（论文）的前置部分包括封面、中英文摘要、目录等。

A.3.1　封面及打印格式

- 学号：按照学校的统一编号，在右上角正确打印自己的学号，宋体，小四号，加粗。
- 题目：题目应和任务书的题目一致，黑体，三号。
- 学院、专业、班级、学生姓名和导师姓名职称等内容，宋体，小三号，居中排列。

A.3.2 中英文摘要及关键词

摘要是关于论文的内容不加注释和评论的简短陈述，具有独立性和自含性。它主要是简要说明研究工作的目的、方法、结果和结论，重点说明本论文的成果和新见解。关键词是为了文献标引工作从论文中选取出来用以表示全文主题内容信息的术语。

1. 中文摘要，宋体小四号，一般为 300 字；英文摘要，“Times News Roman’’ 字体，小四号，一般为 300 个实词。摘要中不宜出现公式、非公用的符号、术语等。

2. 每篇论文选取 3 ~ 5 个关键词，中文为黑体小四号，英文为 “Times News Roman’’ 字体加粗，小四号。关键词排列在摘要的左下方一行，起始格式为：“**关键词**：’’ 和 “**Keyword:**’’。具体的各个关键词以均匀间隔排列，之间不加任何分隔符号。

A.4 目录

按照论文的章、节、附录等前后顺序，编写序号、名称和页码。目录页排在中英文摘要之后，主体部分必须另页右面开始，全文以右页为单页页码。

A.5 毕业设计（论文）的主体部分

毕业设计（论文）的主体部分包括引言（绪论)、正文、结论、结束语、致谢、参考文献。

A.5.1 绪论

作为论文的开端，简要说明作者所做工作的目的、范围、国内外进展情况、前人研究成果、本人的设想、研究方法等。

A.5.2 正文

为毕业设计（论文）的核心部分，包括理论分析、数据资料、实验方法、结果、本人的论点和结论等内容，还要附有各种有关的图表、照片、公式等。要求理论正确、逻辑清楚、层次分明、文字流畅、数据真实可靠，公式推导和计算结果无误，图表绘制要少而精。

图 包括曲线图、示意图、流程图、框图等。图序号一律用阿拉伯数字分章依序编码，如：图 1.3、图 2.11。每一图应有简短确切的图名，连同图序号置于图的正下方。图中坐标上标注的符号和缩略词必须与正文中一致。

表 包括分类项目和数据，一般要求分类项目由左至右横排，数据从上到下竖列。分类项目横排中必须标明符号或单位，竖列的数据栏中不宜出现 ‘‘同上’’、‘‘同左’’ 等类似词语，一律填写具体的数字或文字。表序号一律用阿拉伯数字分章依序编码，如：表 2.5、表 10.3。每一表应有简短确切的题名，连同表序号置于表的正上方。

公式 正文中的公式、算式、方程式等必须编排序号，序号一律用阿拉伯数字分章依序编码，如：式 (3-32)、式 (6-21)。对于较长的公式，另行居中横排，只可在符号处（如：+、-、*、/、<、> 等）转行。公式序号标注于该式所在行 (当有续行时，应标注于最后一行）的最右边。连续性的公式在 “=’’ 处排列整齐。大于 999 的整数或多于三位的小数，一律用半个阿拉伯数字符的小间隔分开；小于 1 的数应将 0 置于小数点之前。

计量单位 单位名称和符号的书写方式一律采用国际通用符号。

A.5.3　结论

是对主体的最终结论，应准确、完整、精炼。阐述作者创造性工作在本研究领域的地位和作用，对存在的问题和不足应给予客观的说明，也可提出进一步的设想。

A.5.4　致谢

对协助完成论文研究工作的单位和个人表示感谢。

A.5.5　参考文献

在学位论文中引用参考文献时，引出处右上角用方括号标注阿拉伯数字编排的序号（必须与参考文献一致)。参考文献的排列格式分为:

专著类的文献 [序号] 作者. 专著名称. 版本. 出版地：出版者，出版年. 参考的页码。

期刊类的文献 作者. 文献名. 期刊名称. 年, 月, 卷（期）. 页码。

其中作者采用姓在前、名在后的形式。当作者超过三个时，只著录前三个人，其后加 ‘‘等’’ 字即可。

A.6　毕业设计（论文）的附录部分

附录是作为学位论文主体的补充，包括下列内容:

1. 正文中过于冗长的公式推导;

2. 为读者阅读方便所需要的辅助性的数学工作或带有重复性的图表；

3. 由于过分冗长而不宜在正文中出现的计算机程序清单；

4. 对于一般读者并非必要阅读，但对本专业同行有参考价值的资料。

5. 附录编于正文后，与正文连续编页码，每一附录均另页起。

6. 附录依次用大写正体 A，B，C……编序号，黑体，三号。如：附录 A。

7. 附录中的图、表、式、参考文献等与正文分开，用阿拉伯数字另行编序号，注意在数码前冠以附录的序码。如：图 A1；表 B2；式（C-3）；文献 [D5]。

A.7 毕业设计（论文）的打印规格

论文正文页面和版面的设置规格：论文正文双面打印，为了便于装订、复制，要求每页纸的四周留有足够的空白边缘。以 WORD97 为例：

页面设置数据为：上 3 厘米、下 2 厘米、内侧 3 厘米、外侧 2 厘米；装订线 – 1 厘米；页眉 - 2 厘米；页脚 - 1 厘米。

版面设置数据为：文字的行间距 - 1. 5 倍；公式的行间距 - 1. 5 倍字符间距 - 标准；页码数据－对称页边距。

A.8 毕业设计（论文）的装订说明

毕业设计（论文）要求以 A4 纸的标准，按照下列顺序装订。外文资料翻译原文及译文另册装订，格式参照论文对应内容格式要求。

1. 封面

2. 任务书

3. 工作计划

4. 中期检查表

5. 成绩考核登记表

6. 中、外论文摘要

7. 目录

8. 引言

9. 论文

10. 结论

11. 结束语

12. 参考文献

13. 附录

致　谢

毕业论文暂告收尾，这也意味着我在西安电子科技大学的四年的学习生活既将结束。回首既往，自己一生最宝贵的时光能于这样的校园之中，能在众多学富五车、才华横溢的老师们的熏陶下度过，实是荣幸之极。在这四年的时间里，我在学习上和思想上都受益非浅。这除了自身努力外，与各位老师、同学和朋友的关心、支持和鼓励是分不开的论文的写作是枯燥艰辛而又富有挑战的。

数学是理论界一直探讨的热门话题，老师的谆谆诱导、同学的出谋划策及家长的支持鼓励，是我坚持完成论文的动力源泉。在此，我特别要感谢我的导师 xxx 老师。从论文的选题、文献的采集、框架的设计、结构的布局到最终的论文定稿，从内容到格式，从标题到标点，她都费尽心血。没有 xxx 老师的辛勤栽培、孜孜教诲，就没有我论文的顺利完成。

感谢数学系的各位同学，与他们的交流使我受益颇多。最后要感谢我的家人以及我的朋友们对我的理解、支持、鼓励和帮助，正是因为有了他们，我所做的一切才更有意义；也正是因为有了他们，我才有了追求进步的勇气和信心。

时间的仓促及自身专业水平的不足，整篇论文肯定存在尚未发现的缺点和错误。恳请阅读此篇论文的老师、同学，多予指正，不胜感激！.....

谨把本文献给我最敬爱的父母亲以及所有关心我的人！

参考文献

[1] Paolo Ferragina and Giovanni Manzini. Indexing compressed text. *Journal of the ACM (JACM)*, 52(4):552–581, 2005.

[2] Roberto Grossi and Jeffrey Scott Vitter. Compressed suffix arrays and suffix trees with applications to text indexing and string matching. *SIAM Journal on Computing*, 35(2):378–407, 2005.

[3] Hongwei Huo, Longgang Chen, Jeffrey Scott Vitter, and Yakov Nekrich. A practical implementation of compressed suffix arrays with applications to self-indexing. In *Data Compression Conference (DCC), 2014*, pages 292–301. IEEE, 2014.

[4] Guy Jacobson. Space-efficient static trees and graphs. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 549–554. IEEE, 1989.

[5] Ben Langmead, Cole Trapnell, Mihai Pop, Steven L Salzberg, et al. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biol*, 10(3):R25, 2009.

[6] Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[7] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, et al. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[8] Heng Li, Jue Ruan, and Richard Durbin. Mapping short dna sequencing reads and calling variants using mapping quality scores. *Genome research*, 18(11):1851–1858, 2008.

[9] Ruiqiang Li, Yingrui Li, Karsten Kristiansen, and Jun Wang. Soap: short oligonucleotide alignment program. *Bioinformatics*, 24(5):713–714, 2008.

[10] Ruiqiang Li, Chang Yu, Yingrui Li, Tak-Wah Lam, Siu-Ming Yiu, Karsten Kristiansen, and Jun Wang. Soap2: an improved ultrafast tool for short read alignment. *Bioinformatics*, 25(15):1966–1967, 2009.

[11] Patel JM Li Y, Terrell A. Wham: a high-throughput sequence alignment method.

[12] Hao Lin, Zefeng Zhang, Michael Q Zhang, Bin Ma, and Ming Li. Zoom! zillions of oligos mapped. *Bioinformatics*, 24(21):2431–2437, 2008.

[13] Ross A Lippert. Space-efficient whole genome comparisons with burrows-wheeler transforms. *Journal of Computational Biology*, 12(4):407–415, 2005.

[14] Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *siam Journal on Computing*, 22(5):935–948, 1993.

[15] Michael L Metzker. Sequencing technologies—the next generation. *Nature Reviews Genetics*, 11(1):31–46, 2009.

[16] Rajeev Raman, Venkatesh Raman, and S Srinivasa Rao. Succinct indexable dictionaries with applications to encoding k-ary trees and multisets. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 233–242. Society for Industrial and Applied Mathematics, 2002.

[17] Stephen M Rumble, Phil Lacroute, Adrian V Dalca, Marc Fiume, Arend Sidow, and Michael Brudno. Shrimp: accurate mapping of short color-space reads. *PLoS computational biology*, 5(5):e1000386, 2009.