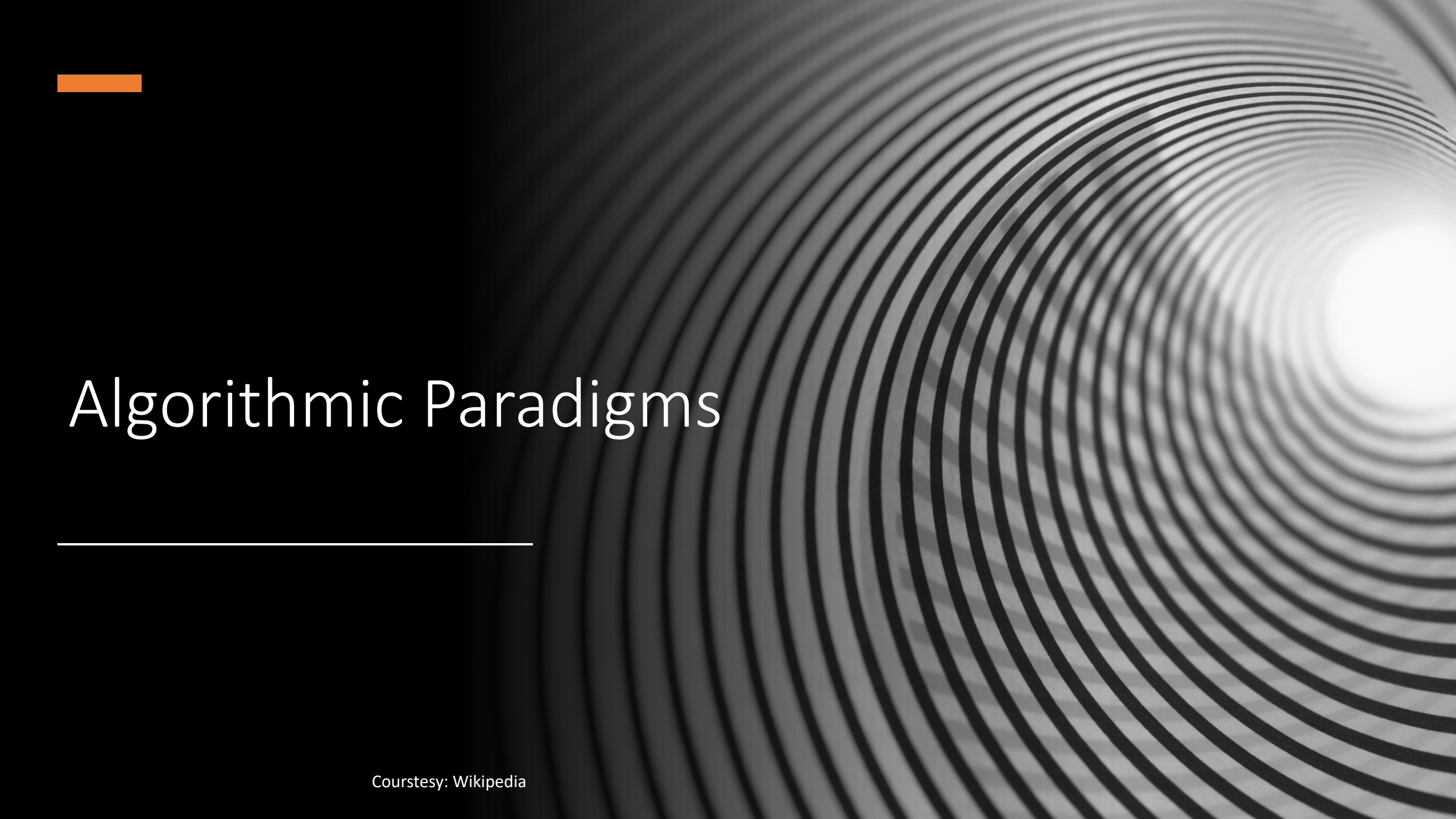
Algorithmic Paradigms



Courstesy: Wikipedia

Protein Sequence FASTA format

>MCHU - Calmodulin - Human, rabbit, bovine, rat, and chicken
MADQLTEEQIAEFKEAFSLFDKDGDGTITT
KELGTVMRSLGQNPTEAELQDMINEVDAD
GNGTID
FPEFLTMMARKMKDTDSEEEIREAFRVFDK
DGNGYISAAELRHVMTNLGEKLTDEEVDE
MIREA
DIDGDGQVNYEEFVQMMTAK*

Courstesy: Wikipedia

DNA Sequence FASTA format

>OV986001.1 Pseudomonas fluorescens SBW25 genome assembly, chromosome: 1

GTGTCAGTGGAACTTTGGCAGCAGTGCG
TGGAGCTTTTGCGCGATGAGCTGCCTGCC
CAGCAATTCAACA
CCTGGATCCGTCCACTACAGGTCGAAGCC
GAAGGCGACGAGTTGCGTGTTTACGCGC
CCAATCGTTTTGT
TCTCGACTGGGTCAACGAGAAGTACCTGA
GCCGCGTGCTCGAATTGCTCGATGAACAC
GGCAACGGCCTC

Courstesy: NCBI

DNA Sequence

GTGTCAGTGGAACTTTGGCAGCAGTGCG
TGGAGCTTTTGCGCGATGAGCTGCCTGCC
CAGCAATTCAACA
CCTGGATCCGTCCACTACAGGTCGAAGCC
GAAGGCGACGAGTTGCGTGTTTACGCGC
CCAATCGTTTTGT
TCTCGACTGGGTCAACGAGAAGTACCTGA
GCCGCGTGCTCGAATTGCTCGATGAACAC
GGCAACGGCCTC

Courstesy: NCBI

Strings in Biology

Protein: Typical length in hundreds, many letters

DNA: Length in millions, 4 letters (A,C,G,T)

Courstesy: NCBI

Algorithmic Problems with Strings

1. Sorting a collection of strings

2. Aligning two strings for similarity

3. Compressing one string for efficient storage

Courstesy: NCBI

Sorting

0: TCAG
1: CGCCT
2: ATT
3: GTA
4: CATG
5: ACGC
6: CCG
7: ACGAT
8: TCAA
9: ATTC

ACGAT
ACGC
ATT
ATTC
CATG
CCG
CGCCT
GTA
TCAA
TCAG

Radix Sort: Recursive

0: TCAG
1: CGCCT
2: ATT
3: GTA
4: CATG
5: ACGC
6: CCG
7: ACGAT
8: TCAA
9: ATTC

ATT
ACGC
ACGAT
ATTC

CGCCT
CATG
CCG

GTA

TCAG
TCAA

Radix Sort: Recursive

ATT
ACGC
ACGAT
ATTC

CGCCT
CATG
CCG

GTA

TCAG
TCAA

Radix Sort: Recursive

ACGAT
ACGC
ATT
ATTC

CGCCT
CATG
CCG

GTA

TCAG
TCAA

Radix Sort: Recursive

ACGAT
ACGC
ATT
ATTC

CATG
CCG
CGCCT

GTA

TCAG
TCAA

Radix Sort: Recursive

ACGAT
ACGC
ATT
ATTC

CATG
CCG
CGCCT

GTA

TCAG
TCAA

Radix Sort: Recursive

ACGAT
ACGC
ATT
ATTC

CATG
CCG
CGCCT

GTA

TCAA
TCAG

Radix Sort: Recursive

Algorithm:

RadixSortRecursive(Strings, position):

1. Create one table for each alphabet symbol.
2. For each string in Strings:
   Store string in Table[string[position]].
3. For each table T,
   T=RadixSortRecursive(T, position+1)
4. Strings=Concatenation of sorted tables.

Return Strings.

Radix Sort: Analysis

Time Complexity:
O(Ln), where L=max length of strings, n=# strings
Space Complexity:
Above Description: O(Ln)

Review Exercise:

Sort the following strings:

CCGA
TT
AATC
GTATG
GGAT
TCCG
CACAGA
AACG
CCG
ACTAG

Radix Sort: Space Optimized

0: TCAG
1: CGCCT
2: ATT
3: GTA
4: CATG
5: ACGC
6: CCG
7: ACGAT
8: TCAA
9: ATTC

ATT
ACGC
ACGAT
ATTC

CGCCT
CATG
CCG

GTA

TCAG
TCAA

Radix Sort: Space Optimized

0: TCAG
1: CGCCT
2: ATT
3: GTA
4: CATG
5: ACGC
6: CCG
7: ACGAT
8: TCAA
9: ATTC

2
5
7
9

1
4
6

3

0
8

Radix Sort: Space Optimized

0: TCAG	2
1: CGCCT	5
2: ATT	7
3: GTA	9
4: CATG	1
5: ACGC	4
6: CCG	6
7: ACGAT	3
8: TCAA	0
9: ATTC	8

2	1
5	4
7	6
9	
3	0
	8

Radix Sort: Space Optimized

0: TCAG	2	--	5
1: CGCCT	5		7
2: ATT	7		
3: GTA	9		
4: CATG	1		
5: ACGC	4		
6: CCG	6		
7: ACGAT	3	--	2
8: TCAA	0		9
9: ATTC	8		

Radix Sort: Space Optimized

0: TCAG	5
1: CGCCT	7
2: ATT	2
3: GTA	9
4: CATG	1
5: ACGC	4
6: CCG	6
7: ACGAT	3
8: TCAA	0
9: ATTC	8

--

5
7

--

2
9

Radix Sort: Iterative

0: TCAG
1: CGCCT
2: ATT
3: GTA
4: CATG
5: ACGC
6: CCG
7: ACGAT
8: TCAA
9: ATTC

ATT
ACGC
ACGAT
ATTC

CGCCT
CATG
CCG

GTA

TCAG
TCAA

Radix Sort: Iterative

0: ATT
1: ACGC
2: ACGAT
3: ATTC
4: CGCCT
5: CATG
6: CCG
7: GTA
8: TCAG
9: TCAA

ATT
ACGC
ACGAT
ATTC

CGCCT
CATG
CCG

GTA

TCAG
TCAA

Radix Sort: Iterative

0: ATT
1: ACGC
2: ACGAT
3: ATTC
4: CGCCT
5: CATG
6: CCG
7: GTA
8: TCAG
9: TCAA

ACGC
ACGAT

ATT
ATTC

Radix Sort: Iterative

0: ACGC
1: ACGAT
2: ATT
3: ATTC
4: CGCCT
5: CATG
6: CCG
7: GTA
8: TCAG
9: TCAA

ACGC
ACGAT

ATT
ATTC

Radix Sort: Iterative

0: ACGC
1: ACGAT
2: ATT
3: ATTC
4: CGCCT
5: CATG
6: CCG
7: GTA
8: TCAG
9: TCAA

CATG CCG

CGCCT ---

Radix Sort: Iterative

0: ACGC
1: ACGAT
2: ATT
3: ATTC
4: CATG
5: CCG
6: CGCCT
7: GTA
8: TCAG
9: TCAA

CATG CCG

CGCCT ---

Radix Sort: Iterative

0: ACGC
1: ACGAT
2: ATT
3: ATTC
4: CATG
5: CCG
6: CGCCT
7: GTA
8: TCAG
9: TCAA

TCAA ---

TCAG ---

Radix Sort: Iterative

0: ACGC
1: ACGAT
2: ATT
3: ATTC
4: CATG
5: CCG
6: CGCCT
7: GTA
8: TCAA
9: TCAG

TCAA ---

TCAG ---

Radix Sort: Iterative

0: TCAA	0: ATT	0: ACGC	0: ACGAT
1: CGCCT	1: ACGC	1: ACGAT	1: ACGC
2: ATT	2: ACGAT	2: ATT	2: ATT
3: GTA	3: ATTC	3: ATTC	3: ATTC
4: CATG	4: CGCCT	4: CATG	4: CATG
5: ACGC	5: CATG	5: CCG	5: CCG
6: CCG	6: CCG	6: CGCCT	6: CGCCT
7: ACGAT	7: GTA	7: GTA	7: GTA
8: TCAG	8: TCAA	8: TCAA	8: TCAG
9: ATTC	9: TCAG	9: TCAG	9: TCAA

Radix Sort: Analysis

Time Complexity:
O(Ln), where L=max length of strings, n=# strings
Space Complexity With Optimization:
O(n+L)
Binary encoding: Zero Additional Space