

FIT2004

Algorithms and Data Structures

Ian Wern Han Lim
lim.wern.han@monash.edu

Referencing materials by
Nathan Companez, Aamir Cheema, Arun Konagurthu and Lloyd Allison



Faculty of Information Technology, Monash University

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

This material has been reproduced and communicated to you by or on behalf of Monash University pursuant to Part VB of the Copyright Act 1968 (the Act). The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act. Do not remove this notice

Ready?

Agenda

- Burrows-Wheeler Transform (BWT)

Agenda

- Burrows-Wheeler Transform (BWT)
 - Why is it awesome?

Agenda

- Burrows-Wheeler Transform (BWT)
 - Why is it awesome?
 - What awesome can you do with it?

Agenda

- Burrows-Wheeler Transform (BWT)
 - Why is it awesome?
 - What awesome can you do with it?
 - How to make it better!

Agenda

- Burrows-Wheeler Transform (BWT)
 - Compression
 - Building the BWT String
 - Substring search with BWT

Let us begin...

Compression

Saving space...

- TL;DR



Compression

Saving space...

- TL;DR
- That is one of the reason why are want compression...
 - There are too much data
 - They are taking up too much space

Compression

Saving space...

- TL;DR
- That is one of the reason why are want compression...
 - There are too much data
 - They are taking up too much space
- Run-length encoding

Compression

Saving space...

- TL;DR
- That is one of the reason why are want compression...
 - There are too much data
 - They are taking up too much space
- Run-length encoding (RLE)
 - Simple lossless data compression

- TL;DR
- That is one of the reason why are want compression...
 - There are too much data
 - They are taking up too much space
- Run-length encoding (RLE)
 - Simple lossless data compression
 - Let us look at an example...

Compression

Saving space...

- “this is mississippi history is this mississippi history”
 - 48 characters, if we ignore the spaces

Compression

Saving space...

- “this is mississippi history is this mississippi history”
 - 48 characters, if we ignore the spaces
 - thisismi2si2si2pihistoryisthismi2si2si2pihistory
 - Still 48 characters...

Compression

Saving space...

- “this is mississippi history is this mississippi history”
 - 48 characters, if we ignore the spaces
 - thisismi2si2si2pihistoryisthismi2si2si2pihistory
 - Still 48 characters...
- “hhhhiiiiiiiiiiiiimmoopprrrrsssssssssssssstttty”
 - 48 characters but sorted now
 - 4h14i2m2o4p2r14s4t2y
 - Only 20 characters!



Compression

Saving space...

- “this is mississippi history is this mississippi history”
 - 48 characters, if we ignore the spaces
 - thisismi2si2si2pihistoryisthismi2si2si2pihistory
 - Still 48 characters...
- “hhhhiiiiiiiiiiiiimmoopprrssssssssssssssttttyy”
 - 48 characters but sorted now
 - 4h14i2m2o4p2r14s4t2y
 - Only 20 characters!
- But sorting is bad...

Compression

Saving space...

- “this is mississippi history is this mississippi history”
 - 48 characters, if we ignore the spaces
 - thisismi2si2si2pihistoryisthismi2si2si2pihistory
 - Still 48 characters...
- “hhhhiiiiiiiiiiiiimmoopprrrrsssssssssssssssttttyy”
 - 48 characters but sorted now
 - 4h14i2m2o4p2r14s4t2y
 - Only 20 characters!
- But sorting is bad...
 - We know the character count
 - But we don’t know the original arrangements of the character

Compression

Saving space...

- “this is mississippi history is this mississippi history”
 - 48 characters, if we ignore the spaces
 - thisismi2si2si2pihistoryisthismi2si2si2pihistory
 - Still 48 characters...

- “hhhhiiiiiiiiiiiiimmoopprrrrssssssssssssssttttyy”
 - 48 characters but sorted now
 - 4h14i2m2o4p2r14s4t2y
 - Only 20 characters!

- But sorting is bad... (also complexity is high)
 - We know the character count
 - But we don’t know the original arrangements of the character

Compression

Saving space...

- So compression that we want
 - Make the original text smaller (we want runs of characters)
 - We can **recover the original string** when we decompress!

Compression

Saving space...

- So compression that we want
 - Make the original text smaller (we want runs of characters)
 - We can **recover the original string** when we decompress!
- So now we will learn burrows-wheeler transform
 - Used in bzip2

Compression

Saving space...

- So compression that we want
 - Make the original text smaller (we want runs of characters)
 - We can **recover the original string** when we decompress!
- So now we will learn burrows-wheeler transform
 - Used in bzip2
- “yttiippsshsssmmyhhssttppiioossissiiiiiii\$sssr”
 - This is the BWT string, we need the \$ for end of string

Compression

Saving space...

- So compression that we want
 - Make the original text smaller (we want runs of characters)
 - We can **recover the original string** when we decompress!
- So now we will learn burrows-wheeler transform
 - Used in bzip2
- “yttiippsshsssmmyhhssttppiioossissiiiiiii\$sssr”
 - This is the BWT string, we need the \$ for end of string
 - y2t2i2p2s2h3s2my2h2s2t2p2i2o2s2i2s9i%3sr
 - 40 characters (we have extra in \$ as well)
 - instead of 48 earlier
 - and is better for real word...

Questions?

Burrow-Wheeler's Transform (BWT)

What is it?

- It is a rearrangement of characters in a string

Burrow-Wheeler's Transform (BWT)

What is it?

- It is a rearrangement of characters in a string
 - \$ usually used to mark the end of the string

Burrow-Wheeler's Transform (BWT)

What is it?

- It is a rearrangement of characters in a string
 - \$ usually used to mark the end of the string
 - Will move the same characters closer
 - That is why we use for compression

Burrow-Wheeler's Transform (BWT)

What is it?

- It is a rearrangement of characters in a string
 - \$ usually used to mark the end of the string
 - Will move the same characters closer
 - That is why we use for compression
 - We can rearrange back to the original string
 - Without needing a key!

Questions?

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
 - String = “apple\$”

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
 - String = “apple\$”
 - apple\$
 - pple\$a
 - ple\$ap
 - le\$app
 - e\$appl
 - \$apple

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
 - String = “apple\$”
 - apple\$
 - pple\$a
 - ple\$ap
 - le\$app
 - e\$appl
 - \$apple
 - We store these strings

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
 - String = “apple\$”
 - apple\$
 - pple\$a
 - ple\$ap
 - le\$app
 - e\$appl
 - \$apple
 - We store these strings
 - What is the complexity?

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
 - String = “apple\$”
 - apple\$
 - pple\$a
 - ple\$ap
 - le\$app
 - e\$appl
 - \$apple
 - We store these strings
 - What is the complexity?
 - $O(N^2)$ time to generate N strings, each of N length
 - $O(N^2)$ space to store them all

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
 - String = “apple\$”
 - apple\$
 - pple\$a
 - ple\$ap
 - le\$app
 - e\$appl
 - \$apple
 - We store these strings
 - What is the complexity?
 - $O(N^2)$ time to generate N strings, each of N length
 - $O(N^2)$ space to store them all
 - But... we can use **suffix arrays**!!! Reducing to $O(N)$

Questions?

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
- Then we sort the strings!

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
- Then we sort the strings!
 - Again, we use suffix array so this can be done in $O(N \log^2 N)$

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
- Then we sort the strings!
 - Again, we use suffix array so this can be done in $O(N \log^2 N)$
- The last column is the BWT string!

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
- Then we sort the strings!
 - Again, we use suffix array so this can be done in $O(N \log^2 N)$
- The last column is the BWT string!
 - We can do this very very fast since the last character is the index position before

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
- Then we sort the strings!
 - Again, we use suffix array so this can be done in $O(N \log^2 N)$
- The last column is the BWT string!
 - We can do this very very fast since the last character is the index position before
 - String = “apple\$”, suffixID = 3 (string from 0), cyclic string = “le\$app”

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
- Then we sort the strings!
 - Again, we use suffix array so this can be done in $O(N \log^2 N)$
- The last column is the BWT string!
 - We can do this very very fast since the last character is the index position before
 - String = “apple\$”, suffixID = 3 (string from 0), cyclic string = “le\$app”
 - What is my last character?

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- Take a string, add \$ behind
- Generate all cyclic representation of a string
- Then we sort the strings!
 - Again, we use suffix array so this can be done in $O(N \log^2 N)$
- The last column is the BWT string!
 - We can do this very very fast since the last character is the index position before
 - String = “apple\$”, suffixID = 3 (string from 0), cyclic string = “le\$app”
 - What is my last character? p which is at index 2
 - Last character = suffix ID - 1

M I S S I S S I P P I \$
 \$ M I S S I S S I P P I
 I \$ M I S S I S S I P P
 P I \$ M I S S I S S I P
 P P I \$ M I S S I S S I
 I P P I \$ M I S S I S S
 S I P P I \$ M I S S I S
 S S I P P I \$ M I S S I
 I S S I P P I \$ M I S S
 S I S S I P P I \$ M I S
 S S I S S I P P I \$ M I
 I S S I S S I P P I \$ M



\$ M I S S I S S I P P I
 I \$ M I S S I S S I P P
 I P P I \$ M I S S I S S
 I S S I P P I \$ M I S S
 I S S I S S I P P I \$ M
 M I S S I S S I P P I \$
 P I \$ M I S S I S S I P
 P P I \$ M I S S I S S I
 S I P P I \$ M I S S I S
 S I S S I P P I \$ M I S
 S S I P P I \$ M I S S I
 S S I S S I P P I \$ M I

All cyclic rotations of the text

Sort the strings in alphabetical order assuming \$ is the smallest

M I S S I S S I P P I \$
 \$ M I S S I S S I P P I
 I \$ M I S S I S S I P P
 P I \$ M I S S I S S I P
 P P I \$ M I S S I S S I
 I P P I \$ M I S S I S S
 S I P P I \$ M I S S I S
 S S I P P I \$ M I S S I
 I S S I P P I \$ M I S S
 S I S S I P P I \$ M I S
 S S I S S I P P I \$ M I
 I S S I S S I P P I \$ M



\$ M I S S I S S I P P I
 I \$ M I S S I S S I P P
 I P P I \$ M I S S I S
 I S S I P P I \$ M I S
 I S S I S S I P P I \$ M
 M I S S I S S I P P I \$
 P I \$ M I S S I S S I P
 P P I \$ M I S S I S S I
 S I P P I \$ M I S S I S
 S I S S I P P I \$ M I S
 S S I P P I \$ M I S S I
 S S I S S I P P I \$ M I

All cyclic rotations of the text

The last column of the sorted matrix is Burrows-Wheeler Transform

Side Note: suffix array uses the first column

M	I	S	S	I	S	S	I	P	P	I	\$
\$	M	I	S	S	I	S	S	I	P	P	I
I	\$	M	I	S	S	I	S	S	I	P	P
P	I	\$	M	I	S	S	I	S	S	I	P
P	P	I	\$	M	I	S	S	I	S	S	I
I	P	P	I	\$	M	I	S	S	I	S	S
S	I	P	P	I	\$	M	I	S	S	I	S
S	S	I	P	P	I	\$	M	I	S	S	I
I	S	S	I	P	P	I	\$	M	I	S	S
S	I	S	S	I	P	P	I	\$	M	I	S
S	S	I	S	S	I	P	P	I	\$	M	I
I	S	S	I	S	S	I	P	P	I	\$	M



Side Note: suffix array uses the first column

\$	M	I	S	S	I	S	S	I	P	P	I
I	\$	M	I	S	S	I	S	S	I	P	P
I	P	P	I	\$	M	I	S	S	I	S	S
I	S	S	I	P	P	I	\$	M	I	S	S
I	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P	I	\$	M	I	S	S	I	S	S	I	P
P	P	I	\$	M	I	S	S	I	S	S	I
S	I	P	P	I	\$	M	I	S	S	I	S
S	I	S	S	I	P	P	I	\$	M	I	S
S	S	I	P	P	I	\$	M	I	S	S	I
S	S	I	S	S	I	P	P	I	\$	M	I

All cyclic rotations of the text

The last column of the sorted matrix is Burrows-Wheeler Transform

M I S S I S S I P P I \$
 \$ M I S S I S S I P P I
 I \$ M I S S I S S I P P
 P I \$ M I S S I S S I P
 P P I \$ M I S S I S S I
 I P P I \$ M I S S I S S
 S I P P I \$ M I S S I S
 S S I P P I \$ M I S S I
 I S S I P P I \$ M I S S
 S I S S I P P I \$ M I S
 S S I S S I P P I \$ M I
 I S S I S S I P P I \$ M



\$ M I S S I S S I P P I
 I \$ M I S S I S S I P P
 I P P I \$ M I S S I S
 I S S I P P I \$ M I S
 I S S I S S I P P I \$ M
 M I S S I S S I P P I \$
 P I \$ M I S S I S S I P
 P P I \$ M I S S I S S I
 S I P P I \$ M I S S I S
 S I S S I P P I \$ M I S
 S S I P P I \$ M I S S I
 S S I S S I P P I \$ M I

All cyclic rotations of the text

The last column of the sorted matrix is Burrows-Wheeler Transform

Questions?

Burrow-Wheeler's Transform (BWT)

So how do we make the string?

- What is the BWT string of “BIRD”?

Burrow-Wheeler's Transform (BWT)

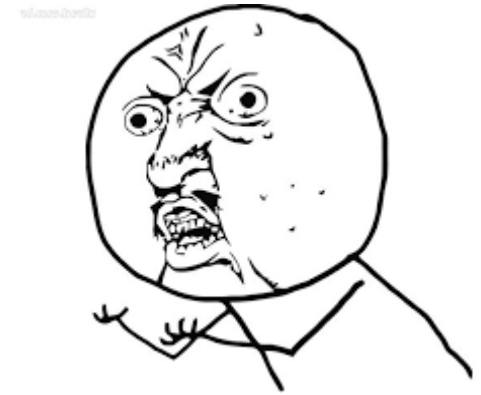
So how do we make the string?

- What is the BWT string of “BIRD”?
 - D\$RBI

Questions?

Burrow-Wheeler's Transform (BWT) Compression

- So why does it work?



Burrow-Wheeler's Transform (BWT)

Compression

- So why does it work?
- It is mainly due to the last-first property from the cyclic rotations



Burrow-Wheeler's Transform (BWT)

Compression

- So why does it work?
- It is mainly due to the last-first property from the cyclic rotations
 - In English, “IS” is very common
 - Thus “I” will occur before “S” more than other letter such as “S” (from “BASS” or “BOSS”)



Burrow-Wheeler's Transform (BWT)

Compression

- So why does it work?
- It is mainly due to the last-first property from the cyclic rotations
 - In English, “IS” is very common
 - Thus “I” will occur before “S” more than other letter such as “S” (from “BASS” or “BOSS”)
 - So once we sort it, then the last column will group similar letter together
 - Such as the “this-is-a-historical-story”



.....
s-a-historical-story\$thi-i
s-is-a-historical-story\$thi
storical-story\$this-is-a-hi
story\$this-is-a-historical-
.....

Questions?

- Given a BWT string, can you reproduce the original string?

- Given a BWT string, can you reproduce the original string?
 - Given “IPSSM\$PISSII”
 - Produce back “MISSISSIPPI\$”

Burrow-Wheeler's Transform (BWT)

Inverting

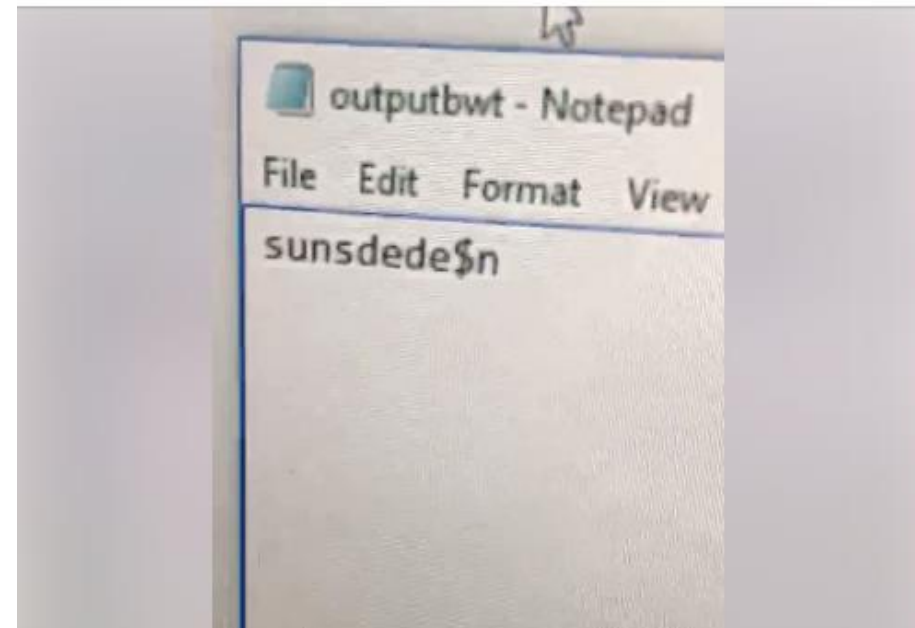
- Given a BWT string, can you reproduce the original string?
 - Given “IPSSM\$PISSII”
 - Produce back “MISSISSIPPI\$”
 - Or from your senior on SoIT Facebook
 - What do you think the original string is?



Alfi Marfaridy

29 April 2017

For those of you who are still struggling with understanding ADS Assignment, hope this video helps with understanding BWT inversion to attain the original string 😊



<https://www.facebook.com/groups/182858515244175/permalink/629405197256169>



Questions?

Burrow-Wheeler's Transform (BWT)

Inverting

- So let us try to inverse “IPSSM\$PISSII”

Burrow-Wheeler's Transform (BWT)

Inverting

- So let us try to inverse “IPSSM\$PISSII”
 - We know this is the last column

- So let us try to inverse “IPSSM\$PISSII”
 - We know this is the last column
 - Can we produce the first column?

- So let us try to inverse “IPSSM\$PISSII”
 - We know this is the last column
 - Can we produce the first column?
 - Yes by sorting! $O(N+M)$ with counting sort, $M = 27$

Burrow-Wheeler's Transform (BWT)

Inverting

- So let us try to inverse “IPSSM\$PISSII”
 - We know this is the last column
 - Can we produce the first column?
 - Yes by sorting! $O(N+M)$ with counting sort, $M = 27$

\$	I
I	P
I	S
I	S
I	M
M	\$
P	P
P	I
S	S
S	S
S	I
S	I

Burrow-Wheeler's Transform (BWT)

Inverting

- So let us try to inverse “IPSSM\$PISSII”
 - We know this is the last column
 - Can we produce the first column?
 - Yes by sorting! $O(N+M)$ with counting sort, $M = 27$
 - Since it is cyclic, we know last column is before first column
 - So we can concatenate both

I \$
P I
S I
S I
M I
\$ M
P P
I P
S S
S S
I S
I S

Burrow-Wheeler's Transform (BWT)

Inverting

- So let us try to inverse “IPSSM\$PISSII”
 - We know this is the last column
 - Can we produce the first column?
 - Yes by sorting! $O(N+M)$ with counting sort, $M = 27$
 - Since it is cyclic, we know last column is before first column
 - So we can concatenate both
 - What happen if we sort it?
 - From LastFirst, we are getting FirstSecond

I \$		\$ M
P I		I \$
S I		I P
S I		I S
M I		I S
\$ M	→ Sort	M I
P P		P I
I P		P P
S S		S I
S S		S I
I S		S S
I S		S S

Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?

Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Start with Last
 - Sort Last to get First

Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Start with Last
 - Sort Last to get First
 - Concatenate Last-First

Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Start with Last
 - Sort Last to get First
 - Concatenate Last-First
 - Sort Last-First to get First-Second

Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Start with Last
 - Sort Last to get First
 - Concatenate Last-First
 - Sort Last-First to get First-Second
 - Concatenate Last-First-Second

Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Start with Last
 - Sort Last to get First
 - Concatenate Last-First
 - Sort Last-First to get First-Second
 - Concatenate Last-First-Second
 - Sort Last-First-Second to get First-Second-Third



Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Start with Last
 - Sort Last to get First
 - Concatenate Last-First
 - Sort Last-First to get First-Second
 - Concatenate Last-First-Second
 - Sort Last-First-Second to get First-Second-Third
 - ...



Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Start with Last
 - Sort Last to get First
 - Concatenate Last-First
 - Sort Last-First to get First-Second
 - Concatenate Last-First-Second
 - Sort Last-First-Second to get First-Second-Third
 - ...
 - ???



Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Start with Last
 - Sort Last to get First
 - Concatenate Last-First
 - Sort Last-First to get First-Second
 - Concatenate Last-First-Second
 - Sort Last-First-Second to get First-Second-Third
 - ...
 - ???
 - Profit!



Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Start with Last
 - Sort Last to get First
 - Concatenate Last-First
 - Sort Last-First to get First-Second
 - Concatenate Last-First-Second
 - Sort Last-First-Second to get First-Second-Third
 - ...
 - ???
 - Profit!

- Let us do it by hand!



Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Start with Last
 - Sort Last to get First
 - Concatenate Last-First
 - Sort Last-First to get First-Second
 - Concatenate Last-First-Second
 - Sort Last-First-Second to get First-Second-Third
 - ...
 - ???
 - Profit!

- Let us do it by hand! Word = “apple\$”



Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?

- Start with Last
 - Sort Last to get First
 - Concatenate Last-First
 - Sort Last-First to get First-Second
 - Concatenate Last-First-Second
 - Sort Last-First-Second to get First-Second-Third
 - ...
 - ???
 - Profit!
-
- Let us do it by hand! Word = “apple\$” it has a bwt string of “e\$lppa”



Questions?

- So what happened?
 - Let us do it by hand! Word = “apple\$” it has a bwt string of “e\$lppa”
 - Or from Nathan's slides...

Inverting

- So what happened?
 - Let us do it by hand! Word = “apple\$” it has a bwt string of “e\$lppa”
 - Or from Nathan's slides...
 - K-mers, all possible substrings of size k (including cyclic rotation)
 - 2-mers of “APPLE\$” is
 - AP
 - PP
 - PL
 - LE
 - E\$
 - \$A

Inverting

- So what happened?
 - Let us do it by hand! Word = “apple\$” it has a bwt string of “e\$lp\$pa”
 - Or from Nathan's slides...
 - K-mers, all possible substrings of size k (including cyclic rotation)
 - 3-mers of “APPLE\$” is
 - APP
 - PPL
 - PLE
 - LE\$
 - E\$A
 - \$AP

Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Concatenating last and first column give us the 2-mers


\$ M I S S I S S I P P I	I \$
I \$ M I S S I S S I P P	P I
I P P I \$ M I S S I S	S I
I S S I P P I \$ M I S	S I
I S S I S S I P P I \$ M	M I
M I S S I S S I P P I \$	\$ M
P I \$ M I S S I S S I P	P P
P P I \$ M I S S I S S I	I P
S I P P I \$ M I S S I S	S S
S I S S I P P I \$ M I S	S S
S S I P P I \$ M I S S I	I S
S S I S S I P P I \$ M I	I S

Concatenate Last
and First columns

Burrow-Wheeler's Transform (BWT)

Inverting

- So what happened?
 - Concatenating last, first and second column give us the 3-mers

\$M I S S I S S I P P I		I \$ M
I \$ M I S S I S S I P P		P I \$
I P P I \$ M I S S I S S		S I P
I S S I P P I \$ M I S S		S I S
I S S I S S I P P I \$ M		M I S
M I S S I S S I P P I \$	 Concatenate Last and First two columns	\$ M I
P I \$ M I S S I S S I P		P P I
P P I \$ M I S S I S S I	I P P	
S I P P I \$ M I S S I S	S S I	
S I S S I P P I \$ M I S	S S I	
S S I P P I \$ M I S S I	I S S	
S S I S S I P P I \$ M I	I S S	

- So what happened?
 - Concatenating last, first and second column give us the 3-mers
 - And so on...

Questions?

- So what is the complexity of our approach?

- So what is the complexity of our approach?
 - We need to sort N times (for N columns)
 - Each column can have a maximum of N characters

- So what is the complexity of our approach?
 - We need to sort N times (for N columns)
 - Each column can have a maximum of N characters
 - So total complexity is $O(N^3)$ for the entire operation

- So what is the complexity of our approach?
 - We need to sort N times (for N columns)
 - Each column can have a maximum of N characters
 - So total complexity is $O(N^3)$ for the entire operation
 - Radix sort is $O(N^2)$
 - We do this N times, concatenating after

- So what is the complexity of our approach?
 - We need to sort N times (for N columns)
 - Each column can have a maximum of N characters
 - So total complexity is $O(N^3)$ for the entire operation
 - Radix sort is $O(N^2)$
 - We do this N times, concatenating after
 - $O(N^2)$ space to store all of the string
 - We can't use suffix array because we dunno the actual word

- So what is the complexity of our approach?
 - We need to sort N times (for N columns)
 - Each column can have a maximum of N characters
 - So total complexity is $O(N^3)$ for the entire operation
 - Radix sort is $O(N^2)$
 - We do this N times, concatenating after
 - $O(N^2)$ space to store all of the string
 - We can't use suffix array because we dunno the actual word

- So what is the complexity of our approach?
 - We need to sort N times (for N columns)
 - Each column can have a maximum of N characters
 - So total complexity is $O(N^3)$ for the entire operation
 - Radix sort is $O(N^2)$
 - We do this N times, concatenating after
 - $O(N^2)$ space to store all of the string
 - We can't use suffix array because we dunno the actual word
- But of course, we do it faster

Questions?

Take a break...

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Let us look at MISSISSIPPI\$ again

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Let us look at MISSISSIPPI\$ again

\$	M	I	S	S	I	S	S	I	P	P	I
I	\$	M	I	S	S	I	S	S	I	P	P
I	P	P	I	\$	M	I	S	S	I	S	S
I	S	S	I	P	P	I	\$	M	I	S	S
I	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P	I	\$	M	I	S	S	I	S	S	I	P
P	P	I	\$	M	I	S	S	I	S	S	I
S	I	P	P	I	\$	M	I	S	S	I	S
S	I	S	S	I	P	P	I	\$	M	I	S
S	S	I	P	P	I	\$	M	I	S	S	I
S	S	I	S	S	I	P	P	I	\$	M	I

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Let us look at MISSISSIPPI\$ again

\$	M	I	S	S	I	S	S	I	P	P	I
I	\$	M	I	S	S	I	S	S	I	P	P
I	P	P	I	\$	M	I	S	S	I	S	S
I	S	S	I	P	P	I	\$	M	I	S	S
I	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P	I	\$	M	I	S	S	I	S	S	I	P
P	P	I	\$	M	I	S	S	I	S	S	I
S	I	P	P	I	\$	M	I	S	S	I	S
S	I	S	S	I	P	P	I	\$	M	I	S
S	S	I	P	P	I	\$	M	I	S	S	I
S	S	I	S	S	I	P	P	I	\$	M	I

← The BWT string

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Let us look at MISSISSIPPI\$ again

\$	M	I	S	S	I	S	S	I	P	P	I	← The original string
I	\$	M	I	S	S	I	S	S	I	P	P	
I	P	P	I	\$	M	I	S	S	I	S	S	
I	S	S	I	P	P	I	\$	M	I	S	S	
I	S	S	I	S	S	I	P	P	I	\$	M	
M	I	S	S	I	S	S	I	P	P	I	\$	
P	I	\$	M	I	S	S	I	S	S	I	P	
P	P	I	\$	M	I	S	S	I	S	S	I	
S	I	P	P	I	\$	M	I	S	S	I	S	
S	I	S	S	I	P	P	I	\$	M	I	S	
S	S	I	P	P	I	\$	M	I	S	S	I	
S	S	I	S	S	I	P	P	I	\$	M	I	

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Let us look at MISSISSIPPI\$ again

\$	M	I	S	S	I	S	S	I	P	P	I
I	\$	M	I	S	S	I	S	S	I	P	P
I	P	P	I	\$	M	I	S	S	I	S	S
I	S	S	I	P	P	I	\$	M	I	S	S
I	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P	I	\$	M	I	S	S	I	S	S	I	P
P	P	I	\$	M	I	S	S	I	S	S	I
S	I	P	P	I	\$	M	I	S	S	I	S
S	I	S	S	I	P	P	I	\$	M	I	S
S	S	I	P	P	I	\$	M	I	S	S	I
S	S	I	S	S	I	P	P	I	\$	M	I

The sorted string

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Look at the "I"s

\$	M	I	S	S	I	S	S	I	P	P	I
I	\$	M	I	S	S	I	S	S	I	P	P
I	P	P	I	\$	M	I	S	S	I	S	S
I	S	S	I	P	P	I	\$	M	I	S	S
I	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P	I	\$	M	I	S	S	I	S	S	I	P
P	P	I	\$	M	I	S	S	I	S	S	I
S	I	P	P	I	\$	M	I	S	S	I	S
S	I	S	S	I	P	P	I	\$	M	I	S
S	S	I	P	P	I	\$	M	I	S	S	I
S	S	I	S	S	I	P	P	I	\$	M	I

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Look at the "I"s

\$	M	I	S	S	I	S	S	I	P	P	I
I	\$	M	I	S	S	I	S	S	I	P	P
I	P	P	I	\$	M	I	S	S	I	S	S
I	S	S	I	P	P	I	\$	M	I	S	S
I	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P	I	\$	M	I	S	S	I	S	S	I	P
P	P	I	\$	M	I	S	S	I	S	S	I
S	I	P	P	I	\$	M	I	S	S	I	S
S	I	S	S	I	P	P	I	\$	M	I	S
S	S	I	P	P	I	\$	M	I	S	S	I
S	S	I	S	S	I	P	P	I	\$	M	I

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Look at the "I"s



\$	M	I	S	S	I	S	S	I	P	P	I
I	\$	M	I	S	S	I	S	S	I	P	P
I	P	P	I	\$	M	I	S	S	I	S	S
I	S	S	I	P	P	I	\$	M	I	S	S
I	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P	I	\$	M	I	S	S	I	S	S	I	P
P	P	I	\$	M	I	S	S	I	S	S	I
S	I	P	P	I	\$	M	I	S	S	I	S
S	I	S	S	I	P	P	I	\$	M	I	S
S	S	I	P	P	I	\$	M	I	S	S	I
S	S	I	S	S	I	P	P	I	\$	M	I

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Look at the "I"s. Their order follows (check with cyclic)...

\$	M	I	S	S	I	S	S	I	P	P	I
I	\$	M	I	S	S	I	S	S	I	P	P
I	P	P	I	\$	M	I	S	S	I	S	S
I	S	S	I	P	P	I	\$	M	I	S	S
I	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P	I	\$	M	I	S	S	I	S	S	I	P
P	P	I	\$	M	I	S	S	I	S	S	I
S	I	P	P	I	\$	M	I	S	S	I	S
S	I	S	S	I	P	P	I	\$	M	I	S
S	S	I	P	P	I	\$	M	I	S	S	I
S	S	I	S	S	I	P	P	I	\$	M	I

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Look at the "S"s. Their order follows too...

\$	M	I	S	S	I	S	S	I	P	P	I
I	\$	M	I	S	S	I	S	S	I	P	P
I	P	P	I	\$	M	I	S	S	I	S	S
I	S	S	I	P	P	I	\$	M	I	S	S
I	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P	I	\$	M	I	S	S	I	S	S	I	P
P	P	I	\$	M	I	S	S	I	S	S	I
S	I	P	P	I	\$	M	I	S	S	I	S
S	I	S	S	I	P	P	I	\$	M	I	S
S	S	I	P	P	I	\$	M	I	S	S	I
S	S	I	S	S	I	P	P	I	\$	M	I

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- If we assign the ID

\$	M	I	S	S	I	S	S	I	P	P	I1
I1	\$	M	I	S	S	I	S	S	I	P	P1
I2	P	P	I	\$	M	I	S	S	I	S	S1
I3	S	S	I	P	P	I	\$	M	I	S	S2
I4	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P1	I	\$	M	I	S	S	I	S	S	I	P2
P2	P	I	\$	M	I	S	S	I	S	S	I2
S1	I	P	P	I	\$	M	I	S	S	I	S3
S2	I	S	S	I	P	P	I	\$	M	I	S4
S3	S	I	P	P	I	\$	M	I	S	S	I3
S4	S	I	S	S	I	P	P	I	\$	M	I4

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- If we assign the ID, the ordering matches!

\$	M	I	S	S	I	S	S	I	P	P	I1
I1	\$	M	I	S	S	I	S	S	I	P	P1
I2	P	P	I	\$	M	I	S	S	I	S	S1
I3	S	S	I	P	P	I	\$	M	I	S	S2
I4	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P1	I	\$	M	I	S	S	I	S	S	I	P2
P2	P	I	\$	M	I	S	S	I	S	S	I2
S1	I	P	P	I	\$	M	I	S	S	I	S3
S2	I	S	S	I	P	P	I	\$	M	I	S4
S3	S	I	P	P	I	\$	M	I	S	S	I3
S4	S	I	S	S	I	P	P	I	\$	M	I4

Faster Inversion of BWT

\$ M I S S I S S I P P I
I \$ M I S S I S S I P P
I P P I \$ M I S S I S S
I S S I P P I \$ M I S S
I S S I S S I P P I \$ M
M I S S I S S I P P I \$
P I \$ M I S S I S S I P
P P I \$ M I S S I S S I
S I P P I \$ M I S S I S
S I S S I P P I \$ M I S
S S I P P I \$ M I S S I
S S I S S I P P I \$ M I

Why does this observation hold?

- Rotate each row that ends at S by one character
- First characters of all these are the same (i.e., S)
- This means the sorting is based on the remaining characters, i.e., the sorting order is determined by stripping off S.
- Hence, the row that appeared earlier before rotation must appear earlier after rotation.

S I P P I \$ M I S S I S
S I S S I P P I \$ M I S
S S I P P I \$ M I S S I
S S I S S I P P I \$ M I

Questions?

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- If we assign the ID, the ordering matches!
 - So do we need the first column?

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- If we assign the ID, the ordering matches!
 - So do we need the first column?
 - No we don't need

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- If we assign the ID, the ordering matches!
 - So do we need the first column?
 - No we don't need
 - All we need is the rank...

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

Rank table

\$	M	I	S	S	I	S	S	I	P	P	I1
I1	\$	M	I	S	S	I	S	S	I	P	P1
I2	P	P	I	\$	M	I	S	S	I	S	S1
I3	S	S	I	P	P	I	\$	M	I	S	S2
I4	S	S	I	S	S	I	P	P	I	\$	M
M	I	S	S	I	S	S	I	P	P	I	\$
P1	I	\$	M	I	S	S	I	S	S	I	P2
P2	P	I	\$	M	I	S	S	I	S	S	I2
S1	I	P	P	I	\$	M	I	S	S	I	S3
S2	I	S	S	I	P	P	I	\$	M	I	S4
S3	S	I	P	P	I	\$	M	I	S	S	I3
S4	S	I	S	S	I	P	P	I	\$	M	I4

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

Rank table

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

Rank table

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S3
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Rank	Char
1	\$
2	I
6	M
7	P
9	S

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Rank table
 - We can build this very quickly by just going through the BWT string
 - Counting the frequency
 - Using the frequency to build the rank

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Rank table
 - We can build this very quickly by just going through the BWT string
 - Counting the frequency
 - Using the frequency to build the rank
 - Done in $O(N)$ time complexity

Questions?

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- How we inverse? “\$”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Rank	Char
1	\$
2	I
6	M
7	P
9	S

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- How we inverse? "\$"

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Rank	Char
1	\$
2	I
6	M
7	P
9	S

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- How we inverse? "I\$"

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Rank	Char
1	\$
2	I
6	M
7	P
9	S

$$\begin{aligned}
 &= \text{Rank}("I") + \text{order}("I") - 1 \\
 &= 2 + 1 - 1 \\
 &= \underline{2}
 \end{aligned}$$

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- How we inverse? "I\$"

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Rank	Char
1	\$
2	I
6	M
7	P
9	S

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- How we inverse? "PI\$"

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Rank	Char
1	\$
2	I
6	M
7	P
9	S

$$\begin{aligned}
 &= \text{Rank}(\text{"P"}) + \text{order}(\text{"P"}) - 1 \\
 &= 7 + 1 - 1 \\
 &= \underline{7}
 \end{aligned}$$

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- How we inverse? "PI\$"

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Rank	Char
1	\$
2	I
6	M
7	P
9	S

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- How we inverse? "PPI\$"

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Rank	Char
1	\$
2	I
6	M
7	P
9	S

$$\begin{aligned}
 &= \text{Rank}(\text{"P"}) + \text{order}(\text{"P"}) - 1 \\
 &= 7 + 2 - 1 \\
 &= \underline{8}
 \end{aligned}$$

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- How we inverse? "PPI\$"

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Rank	Char
1	\$
2	I
6	M
7	P
9	S

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- How we inverse? "IPPI\$"

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Rank	Char
1	\$
2	I
6	M
7	P
9	S

$$\begin{aligned}
 &= \text{Rank}("I") + \text{order}("I") - 1 \\
 &= 2 + 2 - 1 \\
 &= \underline{4}
 \end{aligned}$$

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- How we inverse? "IPPI\$" and so on...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Rank	Char
1	\$
2	I
6	M
7	P
9	S

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Eventually we will reach the first letter of the string and then we are finished...
 - Obtaining the original string

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Eventually we will reach the first letter of the string and then we are finished...
 - Obtaining the original string
- Time complexity?

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Eventually we will reach the first letter of the string and then we are finished...
 - Obtaining the original string
- Time complexity?
 - $O(N)$, following our algorithm earlier with math to know the next index instantly!

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Eventually we will reach the first letter of the string and then we are finished...
 - Obtaining the original string
- Time complexity?
 - $O(N)$, following our algorithm earlier with math to know the next index instantly!
- Space complexity?
 - $O(N)$, for the rank and order array

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Eventually we will reach the first letter of the string and then we are finished...
 - Obtaining the original string
- Time complexity?
 - $O(N)$, following our algorithm earlier with math to know the next index instantly!
- Space complexity?
 - $O(N)$, for the rank and order array
- Let us go through by hand...

Burrow-Wheeler's Transform (BWT)

Inverting the fast way...

- Eventually we will reach the first letter of the string and then we are finished...
 - Obtaining the original string
- Time complexity?
 - $O(N)$, following our algorithm earlier with math to know the next index instantly!
- Space complexity?
 - $O(N)$, for the rank and order array
- Let us go through by hand...
 - Original string = "BANANA\$"
 - BWT string = "ANNB\$AA"

Questions?

Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario...

Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario



Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario



Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario



Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario



Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario



Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario
 - We match DNA



Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario
 - We match DNA
 - A lot of criminal
 - Their DNA is compressed



Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario
 - We match DNA
 - A lot of criminal
 - Their DNA is **compressed** with **BWT**!



Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario
 - We match DNA
 - A lot of criminal
 - Their DNA is **compressed** with **BWT**!
 - Then we just search through the BWT quickly



Questions?

Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario...
 - So we are given “IPSSM\$PISSII” as the BWT

Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario...
 - So we are given “IPSSM\$PISSII” as the BWT
 - Search for “SIS” in the original string...
 - Does it exist???

Burrow-Wheeler's Transform (BWT)

Substring search

- Imagine this scenario...
 - So we are given “IPSSM\$PISSII” as the BWT
 - Search for “SIS” in the original string...
 - Does it exist???

Burrow-Wheeler's Transform (BWT)

Substring search

- We only use the last column really...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Search range for "SIS"

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Search range for "SIS"

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Search range for "SIS"
Note the first and
last "S" within range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Search range for "SI\$"
Note the first and
last "S" within range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Search range for "SIS"
Note the first and
last "S" within range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Search range for "SI\$"
Note the first and
last "S" within range
Then update range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

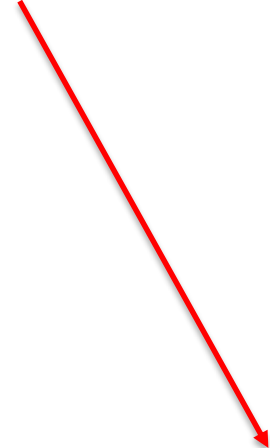
Search range for "SI\$"
Note the first and
last "S" within range
Then update range

Burrow-Wheeler's Transform (BWT)


Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	<u>S1</u>	I	P	P	I	\$	M	I	S	S	I	S3
10	<u>S2</u>	I	S	S	I	P	P	I	\$	M	I	S4
11	<u>S3</u>	S	I	P	P	I	\$	M	I	S	S	I3
12	<u>S4</u>	S	I	S	S	I	P	P	I	\$	M	I4



Search range for "SI\$"
Note the first and
last "S" within range
Then update range

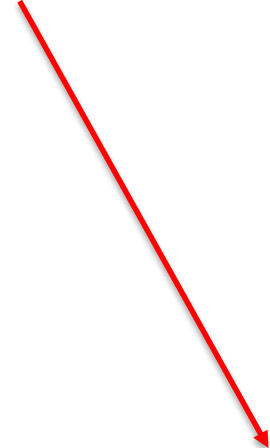


Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Search range for "SIS"
Note the first and
last "I" within range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Search range for "SIS"
Note the first and
last "I" within range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Search range for "SIS"
Note the first and
last "I" within range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Search range for "SIS"
Note the first and
last "I" within range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Search range for "SIS"
Note the first and
last "I" within range
Then update the range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	<u>I3</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S	S2
5	<u>I4</u>	<u>S</u>	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Search range for "SIS"
Note the first and
last "I" within range
Then update the range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



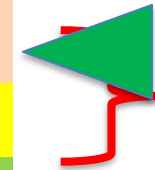
Search range for "SIS"
Note the first and
last "S" within range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	<u>I3</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S	S2
5	<u>I4</u>	<u>S</u>	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



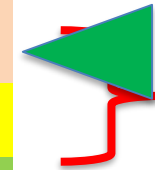
Search range for "SIS"
Note the first and
last "S" within range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	<u>I3</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S	S2
5	<u>I4</u>	<u>S</u>	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



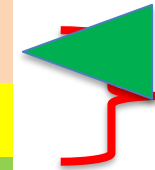
Search range for "SIS"
Note the first and
last "S" within range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	<u>I3</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S	S2
5	<u>I4</u>	<u>S</u>	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



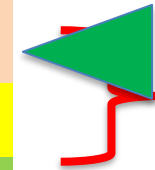
Search range for "SIS"
Note the first and
last "S" within range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	<u>I3</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S	S2
5	<u>I4</u>	<u>S</u>	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Search range for "SIS"
Note the first and
last "S" within range
Then we update range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	<u>I3</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S	S2
5	<u>I4</u>	<u>S</u>	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Search range for "SIS"
Note the first and
last "S" within range
Then we update range

Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We search from the back...

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	<u>I3</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S	S2
5	<u>I4</u>	<u>S</u>	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	<u>S2</u>	<u>I</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Search range for "SIS"
Note the first and
last "S" within range
Then we update range



Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We found our string

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	<u>I3</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S	S2
5	<u>I4</u>	<u>S</u>	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	<u>S2</u>	<u>I</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

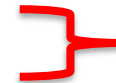


Burrow-Wheeler's Transform (BWT)

Substring search

- Initially, full search range. We found our string

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	<u>I3</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S	S2
5	<u>I4</u>	<u>S</u>	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	<u>S2</u>	<u>I</u>	<u>S</u>	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Questions?

Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for "ISS"

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

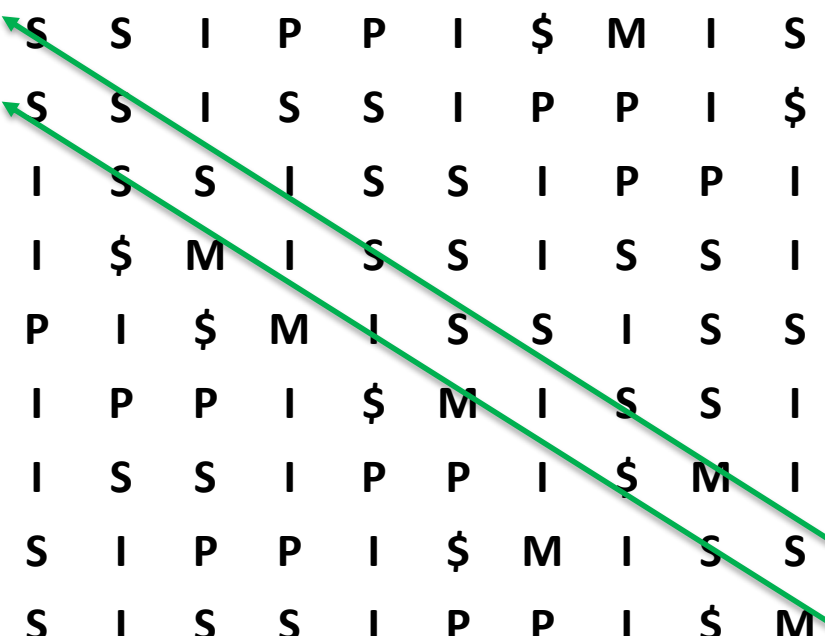



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “ISS”. We found 2 occurrences!

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Questions?

Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “MSS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for "MSS"

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “MSS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “MSS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “MSS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “MSS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4



Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for “MSS”

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

} ...but no M in range!

Burrow-Wheeler's Transform (BWT)

Substring search

- Now try to search for "MSS". MSS doesn't exist!

1	\$	M	I	S	S	I	S	S	I	P	P	I1
2	I1	\$	M	I	S	S	I	S	S	I	P	P1
3	I2	P	P	I	\$	M	I	S	S	I	S	S1
4	I3	S	S	I	P	P	I	\$	M	I	S	S2
5	I4	S	S	I	S	S	I	P	P	I	\$	M
6	M	I	S	S	I	S	S	I	P	P	I	\$
7	P1	I	\$	M	I	S	S	I	S	S	I	P2
8	P2	P	I	\$	M	I	S	S	I	S	S	I2
9	S1	I	P	P	I	\$	M	I	S	S	I	S3
10	S2	I	S	S	I	P	P	I	\$	M	I	S4
11	S3	S	I	P	P	I	\$	M	I	S	S	I3
12	S4	S	I	S	S	I	P	P	I	\$	M	I4

} ...but no M in range!

Questions?

Burrow-Wheeler's Transform (BWT)

Substring search

- Complexity?

Burrow-Wheeler's Transform (BWT)

Substring search

- Complexity?
 - String is N characters
 - Pattern (what we searching) is M character

- Complexity?
 - String is N characters
 - Pattern (what we searching) is M character
 - $O(M \log N)$ using binary search on character position

- Complexity?
 - String is N characters
 - Pattern (what we searching) is M character
 - $O(M \log N)$ using binary search on character position
 - We will go through this more in the tutorial
 - Where the class is smaller and I can be more detailed =)
 - If we do what we just see
 - Then it becomes $O(M)$
 - Using the formula $\text{rank}(\text{char}) + \text{order}(\text{char})$

Questions?

Thank You