COMP9319 Web Data Compression and Search

BWT revisit

Backward Search overview

Recall: Last column = BWT

A	B
A	N
A	N
B	[
N	A
N	A
[	]
]	A
First	Last

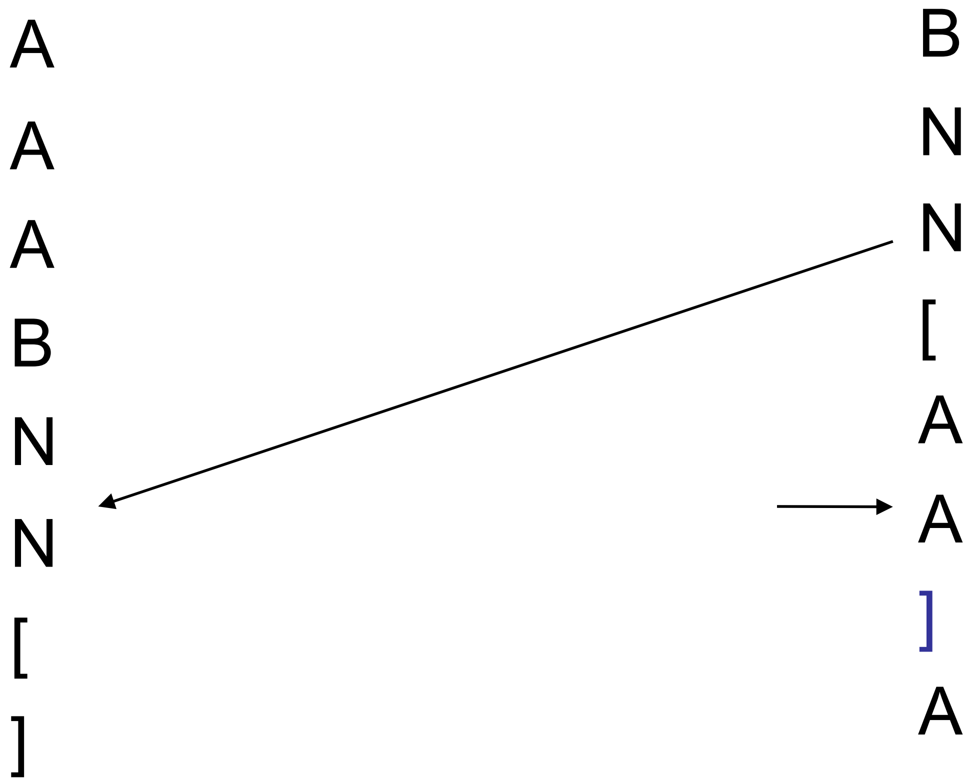
A]

$$\begin{tikzcd}
A & & B \\
A & & N \\
A & & N \\
B & & {[} \\
N & & A \\
N & & A \\
{[} & & {]} \arrow[dll] \\
{]} & {} \arrow[r] & A
\end{tikzcd}$$

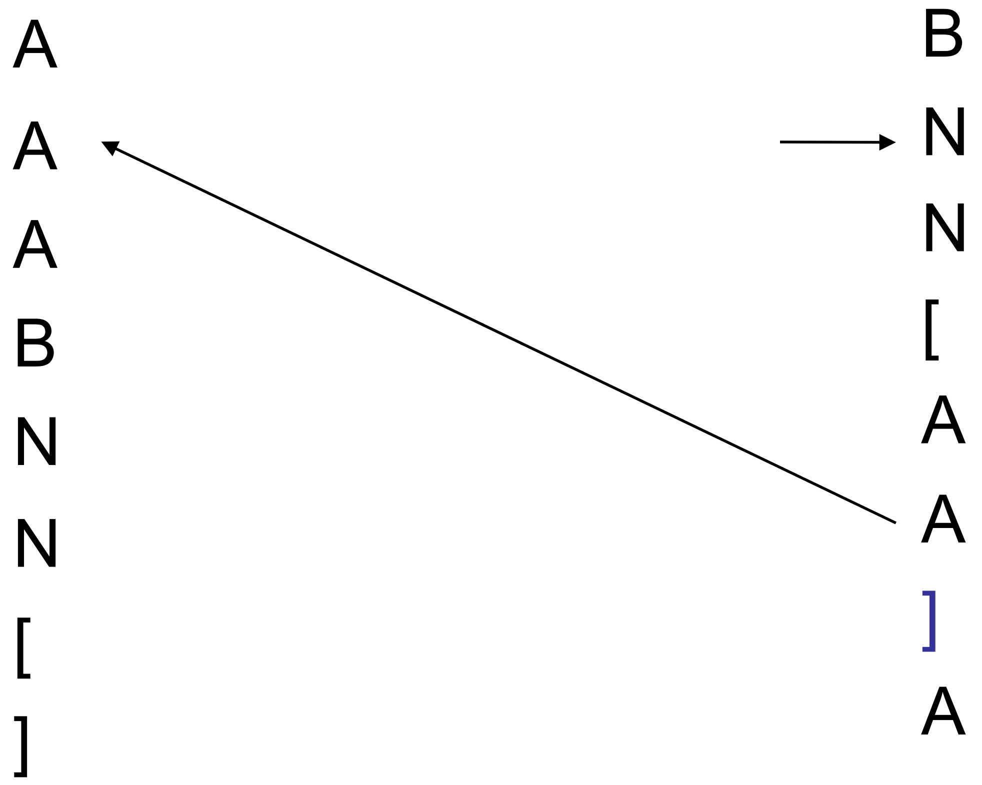
NA]

$$\begin{tikzcd}
-A & & B \\
=A & & N \\
A & {} \arrow[r] & N \\
B & & {[} \\
N & & A- \\
N & & A= \\
{[} & & {]} \\
{]} & & A \arrow[uuuuull]
\end{tikzcd}$$

ANA]



NANA]



ANANA]

$$\begin{tikzcd}
A & & B \\
A & & N \arrow[ddll] \\
A & & N \\
B & & {[} \\
N & {} \arrow[r] & A \\
N & & A \\
{[} & & {]} \\
{]} & & A
\end{tikzcd}$$

BANANA]

$$\begin{tikzcd}
A & {} \arrow[r] & B \\
A & & N \\
A & & N \\
B & & {[} \\
N & & A \arrow[uuuull] \\
N & & A \\
{[} & & {]} \\
{]} & & A
\end{tikzcd}$$

[BANANA]

$$\begin{tikzcd}
A & & B \arrow[dddll] \\
A & & N \\
A & & N \\
B & {} \arrow[r] & {[} \\
N & & A \\
N & & A \\
{[} & & {]} \\
{]} & & A
\end{tikzcd}$$

Example using C[ ] & Occ[ ]

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

???????]

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

??????A]

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

??????NA]

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

????ANA]

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

???NANA]

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

??ANANA]

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

?BANANA]

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

[BANANA]

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

[BANANA]

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Occ / Rank

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

C [ ]

C[ ] & Occ()

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Occ(Symbol, Pos)
=> # Matching

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

C[Symbol] =>
(startPos, endPos)

C[ ] & Occ()

C[Symbol] =>
(startPos, endPos)

Occ(Symbol, Pos)
=> # Matching

Can these two functions (or tables) be implemented such that they can return the result in constant time ?
Yes, have a precomputed table.
Can they be precomputed efficiently ?
Yes, a single pass.

Backward Search for $\underleftarrow{\underline{\text{ANA}}}$

A	B
A	N
A	N
B	[
N	A
N	A
[	]
]	A

Backward Search for NAN

$\longleftarrow$

A	B
A	N
A	N
B	[
N	A
N	A
[	]
]	A

Backward Search for ANA

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Occ(Symbol, Pos)
=> # Matching

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

C[Symbol] =>
(startPos, endPos)

Backward Search for NAN

Position	Symbol	# Matching
0	B	0
1	N	0
2	N	1
3	[	0
4	A	0
5	A	1
6	]	0
7	A	2

Occ(Symbol, Pos) => # Matching

Symbol	# LessThan
A	0
B	3
N	4
[	6
]	7

C[Symbol] => (startPos, endPos)

Why not Forward Search: $\underrightarrow{\text{ANA}}$

A

A

A

B

N

N

[

]

B

N

N

[

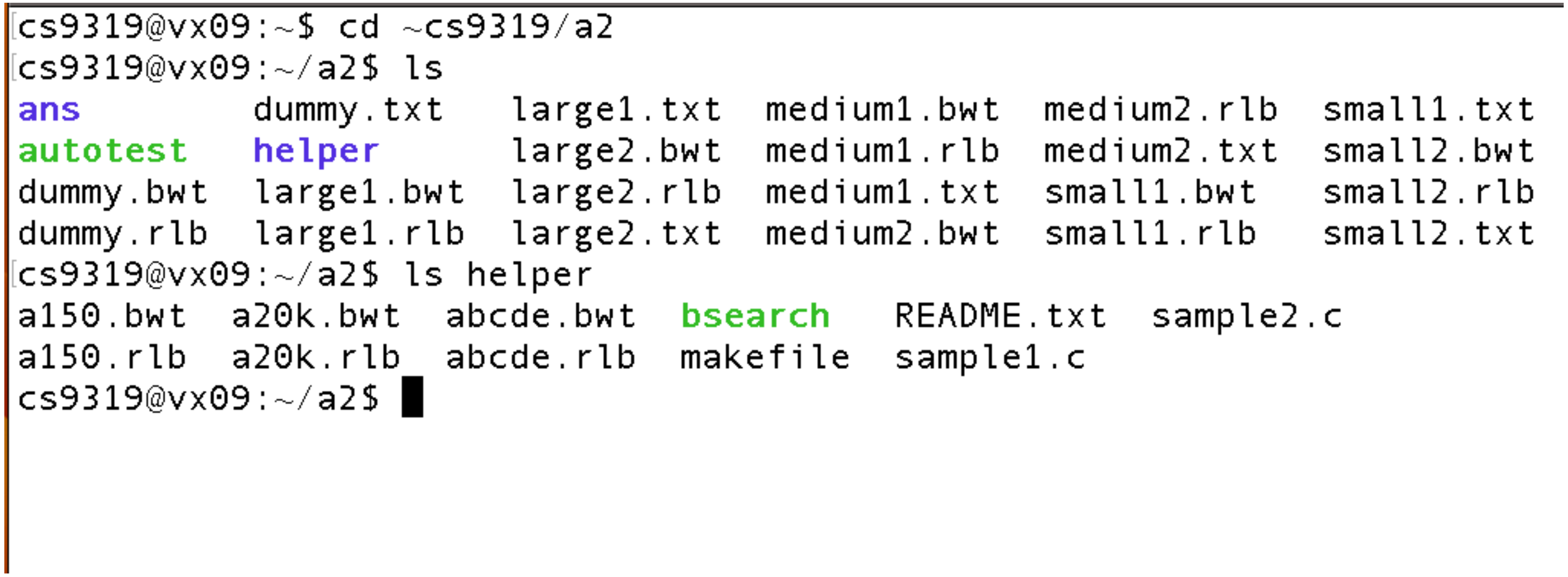
A

A

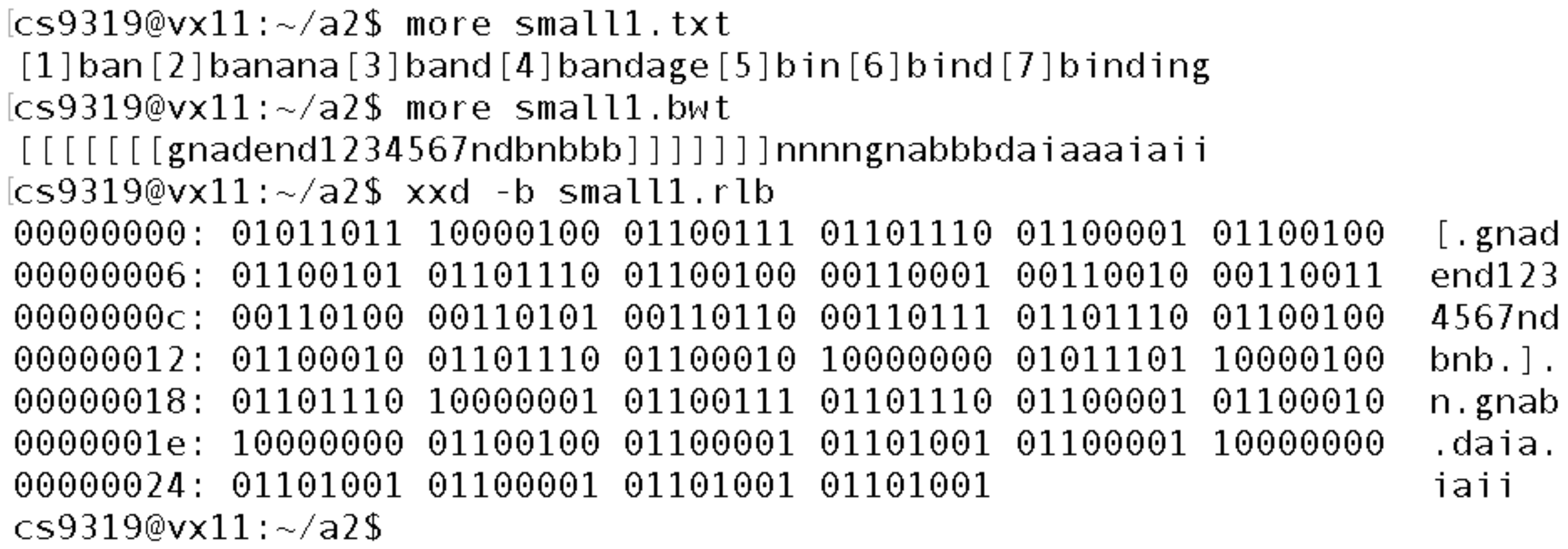
]

A

Assignment 2 overview & tips to start…

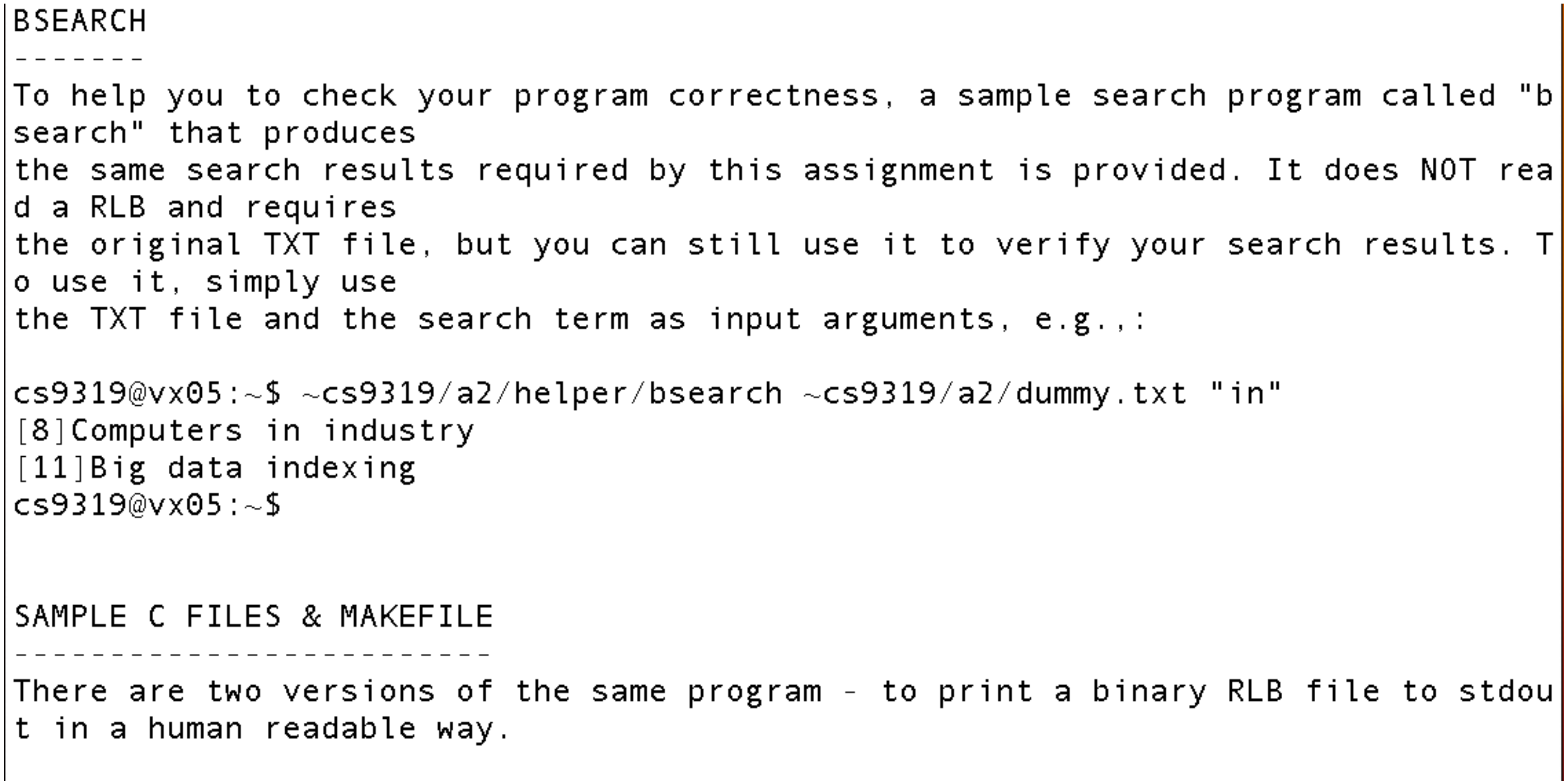


```
cs9319@vx09:~$ cd ~cs9319/a2
cs9319@vx09:~/a2$ ls
ans         dummy.txt   large1.txt  medium1.bwt  medium2.rlb  small1.txt
autotest    helper      large2.bwt  medium1.rlb  medium2.txt  small2.bwt
dummy.bwt   large1.bwt  large2.rlb  medium1.txt  small1.bwt   small2.rlb
dummy.rlb   large1.rlb  large2.txt  medium2.bwt  small1.rlb   small2.txt
cs9319@vx09:~/a2$ ls helper
a150.bwt  a20k.bwt  abcde.bwt  bsearch   README.txt  sample2.c
a150.rlb  a20k.rlb  abcde.rlb  makefile  sample1.c
cs9319@vx09:~/a2$ █
```



```
cs9319@vx11:~/a2$ more small1.txt
 [1]ban[2]banana[3]band[4]bandage[5]bin[6]bind[7]binding
cs9319@vx11:~/a2$ more small1.bwt
 [[[[[[[gnadend1234567ndbnbbb]]]]]]]nnnngnabbbdaiaaaiaii
cs9319@vx11:~/a2$ xxd -b small1.rlb
00000000: 01011011 10000100 01100111 01101110 01100001 01100100  [.gnad
00000006: 01100101 01101110 01100100 00110001 00110010 00110011  end123
0000000c: 00110100 00110101 00110110 00110111 01101110 01100100  4567nd
00000012: 01100010 01101110 01100010 10000000 01011101 10000100  bnb.].
00000018: 01101110 10000001 01100111 01101110 01100001 01100010  n.gnab
0000001e: 10000000 01100100 01100001 01101001 01100001 10000000  .daia.
00000024: 01101001 01100001 01101001 01101001                    iaii
cs9319@vx11:~/a2$
```

README.txt

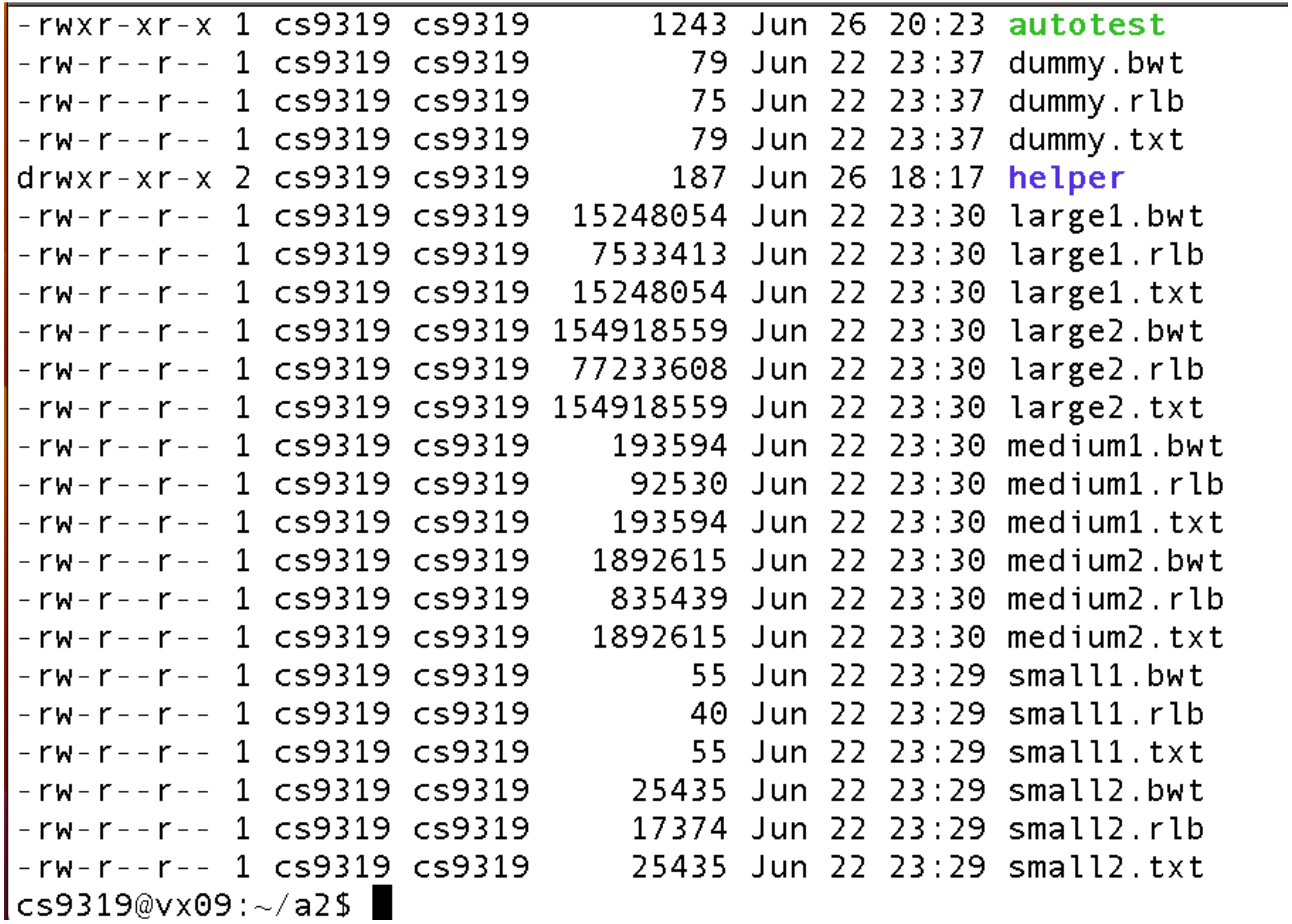


```
BSEARCH
--------
To help you to check your program correctness, a sample search program called "b
search" that produces
the same search results required by this assignment is provided. It does NOT rea
d a RLB and requires
the original TXT file, but you can still use it to verify your search results. T
o use it, simply use
the TXT file and the search term as input arguments, e.g.,:

cs9319@vx05:~$ ~cs9319/a2/helper/bsearch ~cs9319/a2/dummy.txt "in"
[8]Computers in industry
[11]Big data indexing
cs9319@vx05:~$



SAMPLE C FILES & MAKEFILE
-------------------------
There are two versions of the same program - to print a binary RLB file to stdou
t in a human readable way.
```



```
-rwxr-xr-x 1 cs9319 cs9319      1243 Jun 26 20:23 autotest
-rw-r--r-- 1 cs9319 cs9319        79 Jun 22 23:37 dummy.bwt
-rw-r--r-- 1 cs9319 cs9319        75 Jun 22 23:37 dummy.rlb
-rw-r--r-- 1 cs9319 cs9319        79 Jun 22 23:37 dummy.txt
drwxr-xr-x 2 cs9319 cs9319       187 Jun 26 18:17 helper
-rw-r--r-- 1 cs9319 cs9319  15248054 Jun 22 23:30 large1.bwt
-rw-r--r-- 1 cs9319 cs9319   7533413 Jun 22 23:30 large1.rlb
-rw-r--r-- 1 cs9319 cs9319  15248054 Jun 22 23:30 large1.txt
-rw-r--r-- 1 cs9319 cs9319 154918559 Jun 22 23:30 large2.bwt
-rw-r--r-- 1 cs9319 cs9319  77233608 Jun 22 23:30 large2.rlb
-rw-r--r-- 1 cs9319 cs9319 154918559 Jun 22 23:30 large2.txt
-rw-r--r-- 1 cs9319 cs9319    193594 Jun 22 23:30 medium1.bwt
-rw-r--r-- 1 cs9319 cs9319     92530 Jun 22 23:30 medium1.rlb
-rw-r--r-- 1 cs9319 cs9319    193594 Jun 22 23:30 medium1.txt
-rw-r--r-- 1 cs9319 cs9319   1892615 Jun 22 23:30 medium2.bwt
-rw-r--r-- 1 cs9319 cs9319    835439 Jun 22 23:30 medium2.rlb
-rw-r--r-- 1 cs9319 cs9319   1892615 Jun 22 23:30 medium2.txt
-rw-r--r-- 1 cs9319 cs9319        55 Jun 22 23:29 small1.bwt
-rw-r--r-- 1 cs9319 cs9319        40 Jun 22 23:29 small1.rlb
-rw-r--r-- 1 cs9319 cs9319        55 Jun 22 23:29 small1.txt
-rw-r--r-- 1 cs9319 cs9319     25435 Jun 22 23:29 small2.bwt
-rw-r--r-- 1 cs9319 cs9319     17374 Jun 22 23:29 small2.rlb
-rw-r--r-- 1 cs9319 cs9319     25435 Jun 22 23:29 small2.txt
cs9319@vx09:~/a2$ 
```