YEAH Hours

Alexander De Baets

Helpful Resources

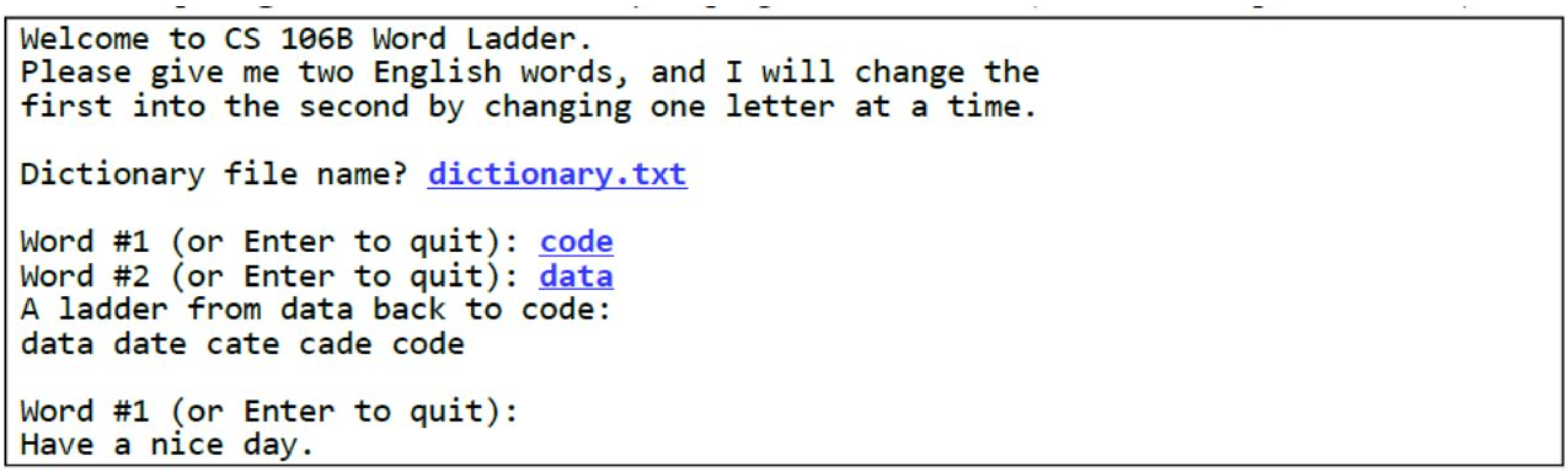
- Lecture 4 ADTs: Stack, Queue
- CS106B+X Style Guide
- Stanford C++ Documentation

Other helpful resources!

- The CLaIR!
    - 8PM-10PM Sunday through Thursday at Old Union
    - This is for conceptual questions.
- The LaIR!
    - 6PM-Midnight Sunday through Thursday at Old Union.
    - This is more geared for students who have started and has bugs!

PROJECT 1: WORD LADDER

Word Ladder!

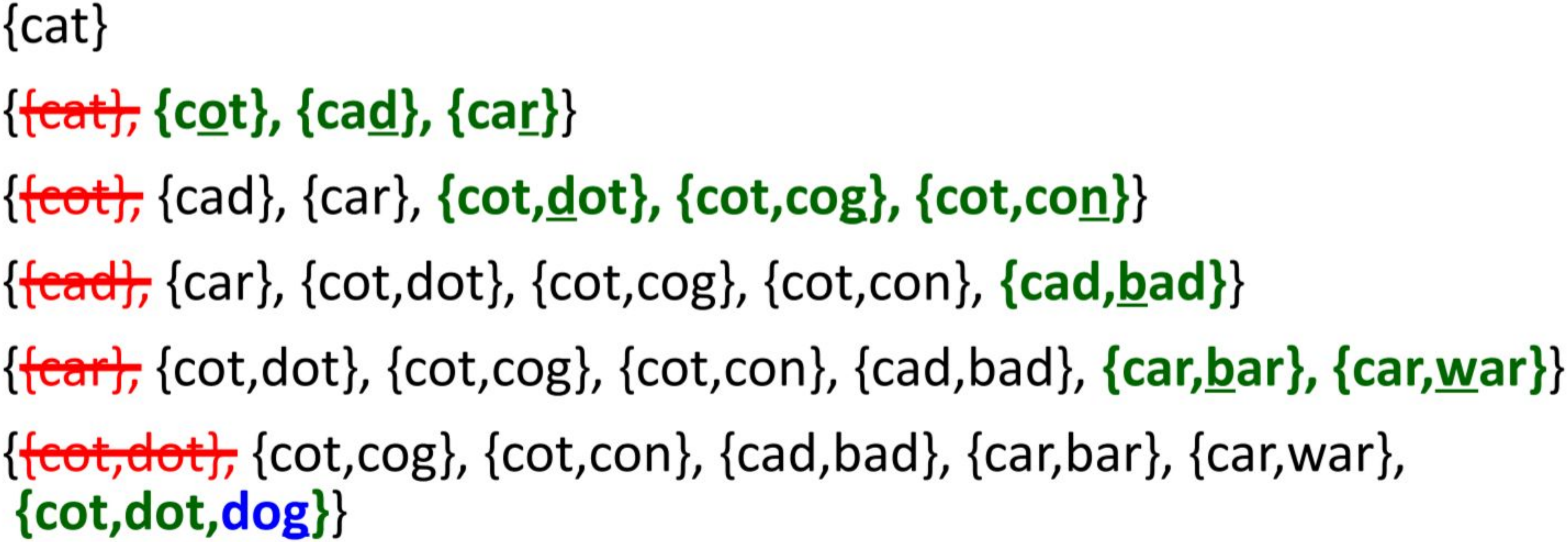


```
Welcome to CS 106B Word Ladder.
Please give me two English words, and I will change the
first into the second by changing one letter at a time.

Dictionary file name? dictionary.txt

Word #1 (or Enter to quit): code
Word #2 (or Enter to quit): data
A ladder from data back to code:
data date cate cade code

Word #1 (or Enter to quit):
Have a nice day.
```

Word Ladder

- First, enter a dictionary (such as dictionary.txt)
- Then, enter in a word. This word MUST be in the dictionary!
- Next, enter in another word. This word must be in the dictionary AND be of the same length of the first word AND not the same as the first word
- Find the word ladder!

Actually finding the word ladder - BFS



{cat}

{~~{cat},~~ {cot}, {cad}, {car}}

{~~{cot},~~ {cad}, {car}, {cot,dot}, {cot,cog}, {cot,con}}

{~~{cad},~~ {car}, {cot,dot}, {cot,cog}, {cot,con}, {cad,bad}}

{~~{car},~~ {cot,dot}, {cot,cog}, {cot,con}, {cad,bad}, {car,bar}, {car,war}}

{~~{cot,dot},~~ {cot,cog}, {cot,con}, {cad,bad}, {car,bar}, {car,war},
{cot,dot,dog}}

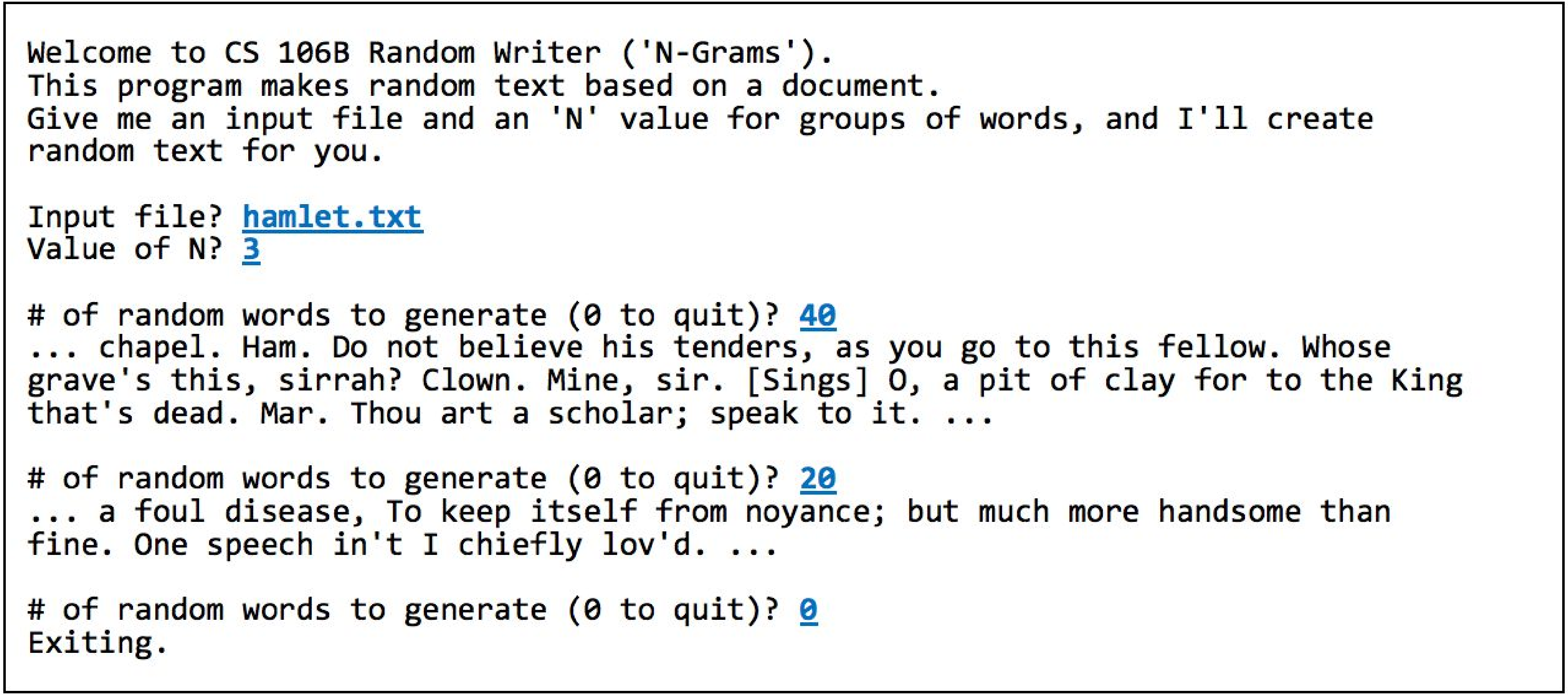
The Pseudo Code

```
create an empty queue of stacks.
create/add a stack containing {w1} to the queue.
while the queue is not empty:
   dequeue partial-ladder stack
   for each valid "neighbor" of the top word:
      if that neighbor hasn't been used before:
         if the neighbor word is w2:
            hooray! we have found a solution.
         otherwise:
            create a copy of the current stack.
            add neighbor word to the copy stack.
            add the copy stack to queue.
```

A couple quick notes

- You can ignore case!
- Pass your lexicon by reference!
- Make a copy of your stack!
- Again, make sure you check all the user inputs!

PROJECT 2: TWO FLAVORS OF RANDOM SENTENCE GENERATOR

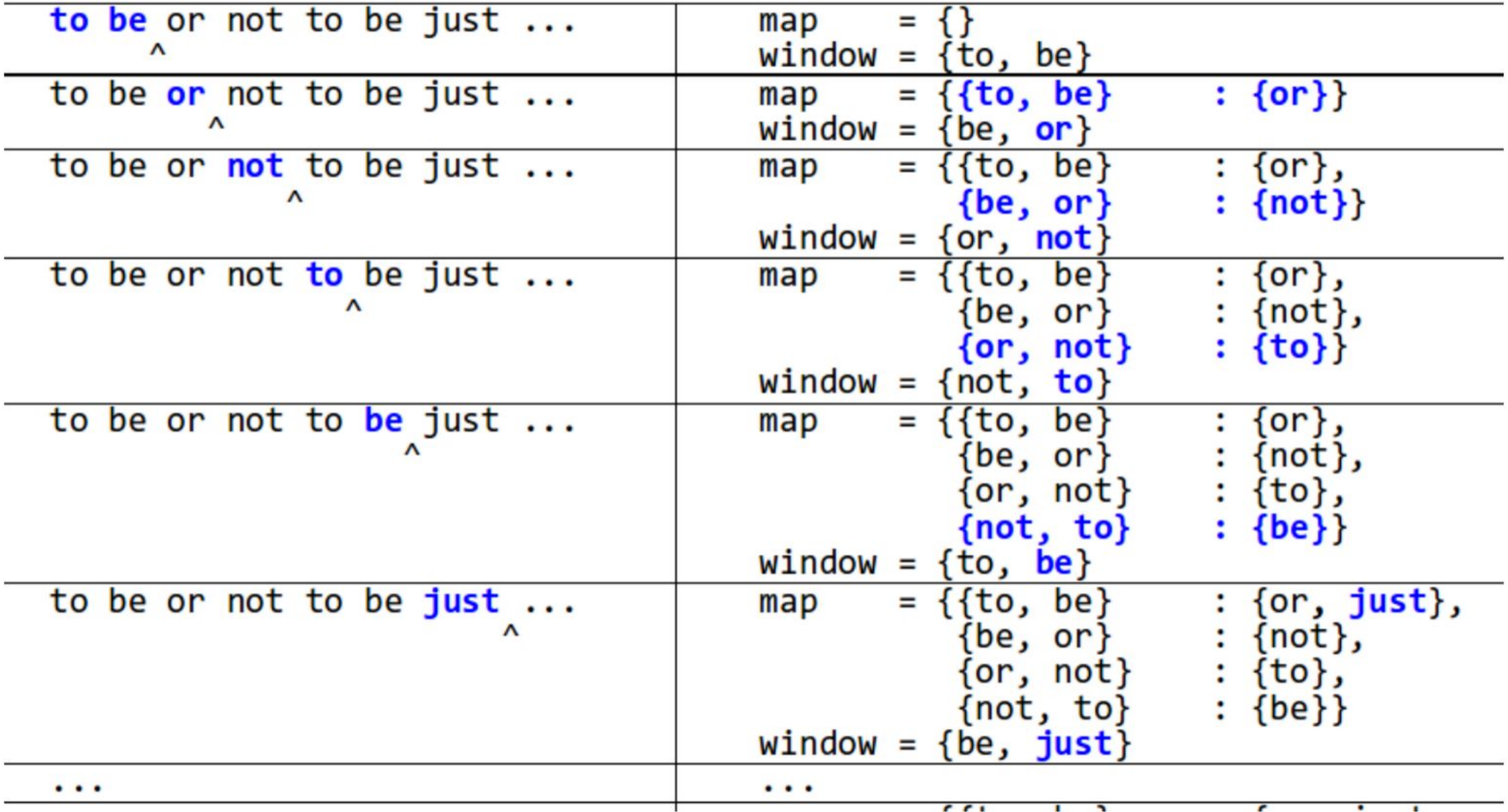


```
Welcome to CS 106B Random Writer ('N-Grams').
This program makes random text based on a document.
Give me an input file and an 'N' value for groups of words, and I'll create
random text for you.

Input file? hamlet.txt
Value of N? 3

# of random words to generate (0 to quit)? 40
... chapel. Ham. Do not believe his tenders, as you go to this fellow. Whose
grave's this, sirrah? Clown. Mine, sir. [Sings] O, a pit of clay for to the King
that's dead. Mar. Thou art a scholar; speak to it. ...

# of random words to generate (0 to quit)? 20
... a foul disease, To keep itself from noyance; but much more handsome than
fine. One speech in't I chiefly lov'd. ...

# of random words to generate (0 to quit)? 0
Exiting.
```

Random Sentence Generators

- First, prompt the user for the file (such as Hamlet.txt)
- Prompt the user for the N value (more on this in a second!)
- Prompt the user for the number of random words to generate!

Step 1: Create the Map!



to be or not to be just ...
^
	map = {}
	window = {to, be}

to be **or** not to be just ...
^
	map = {**{to, be} : {or}**}
	window = {be, **or**}

to be or **not** to be just ...
^
	map = {{to, be} : {or},
	{be, or} : {not}}
	window = {or, **not**}

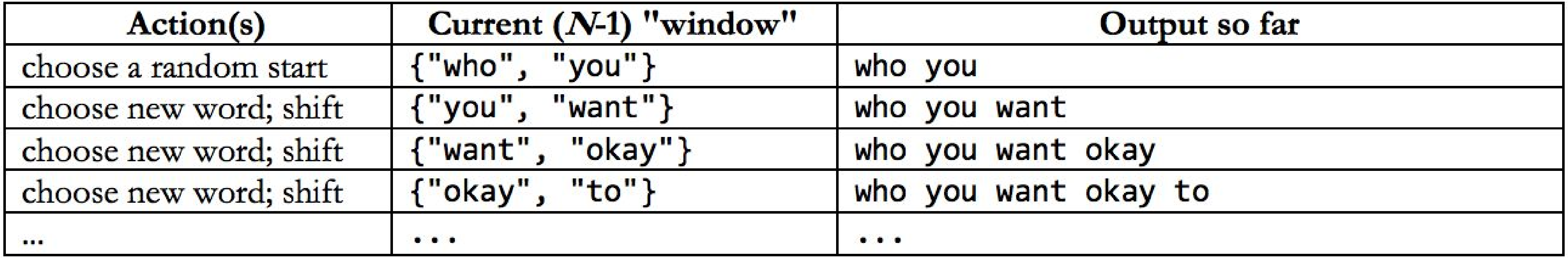
to be or not **to** be just ...
^
	map = {{to, be} : {or},
	{be, or} : {not},
	{or, not} : {to}}
	window = {not, **to**}

to be or not to **be** just ...
^
	map = {{to, be} : {or},
	{be, or} : {not},
	{or, not} : {to},
	{not, to} : {be}}
	window = {to, **be**}

to be or not to be **just** ...
^
	map = {{to, be} : {or, **just**},
	{be, or} : {not},
	{or, not} : {to},
	{not, to} : {be}}
	window = {be, **just**}

...	...

Step 2: Generate Text!



Action(s)	Current (*N*-1) "window"	Output so far
choose a random start	`{"who", "you"}`	`who you`
choose new word; shift	`{"you", "want"}`	`who you want`
choose new word; shift	`{"want", "okay"}`	`who you want okay`
choose new word; shift	`{"okay", "to"}`	`who you want okay to`
...	`...`	`...`

Good Luck!