De Bruijn Graph text reconstruction

- Choose any kmer to start. I choose the first one, aidto
- Start a new node and mark the kmer as used (I use a fill color), Do not delete the word, you'll need it later
- Start a new merged node
- Find the next node by looking up the last four letters of the current word in the list (idto) in the list – or use the find function (easier)
- Add successive nodes to the merged node until you reach a branch point (two or more words overlap)

1			Debruin Graph nodes				
2		kmers	incoming edge	node number	merged node	outgoing edge	
3		aidto		0	aidtobe		
4		amlet					
5		atist					
6		beorn					
7		betha					
8		dtobe					
9		eorno					
10		eques					
11		ethat					
12		etsai					
13		hamle					
14		hatis					
15		heque					
16		idtob					
17		isthe					
18		letsa					
19		mlets					
20		notto					
21		obeor					
22		obeth					
23		ornot					
24		ottob					
25		rnott					
26		saidt					
27		stheq					
28		thati					
29		thequ					
30		tisth					
31		tobeo					
32		tobet					
33		tsaid					
34		ttobe					

De Bruijn Graph text reconstruction

- At this point there is a branch, two words beginning tobe.
- Create two new merged nodes
- Mark nodes 1 and 2 as the outgoing edges of node 0
- Mark node 0 as the incoming edge of nodes 1 and 2
- Choose one of the new nodes to continue working on, node 0 is complete on the right hand side

2		Debruin Graph nodes			
3	kmers	incoming edge	node number	merged node	outgoing edge
4	aidto		0	aidtobe	
5	amlet				
6	atist				
7	beorn				
8	betha				
9	dtobe				
10	eorno				
11	eques				
12	ethat				
13	etsai				
14	hamle				
15	hatis				
16	heque				
17	idtob				
18	isthe				
19	letsa				
20	mlets				
21	notto				
22	obeor				
23	obeth				
24	ornot				
25	ottob				
26	rnott				
27	saidt				
28	stheq				
29	thati				
30	thequ				
31	tisth				
32	tobeo				
33	tobet				
34	tsaid				
35	ttobe				

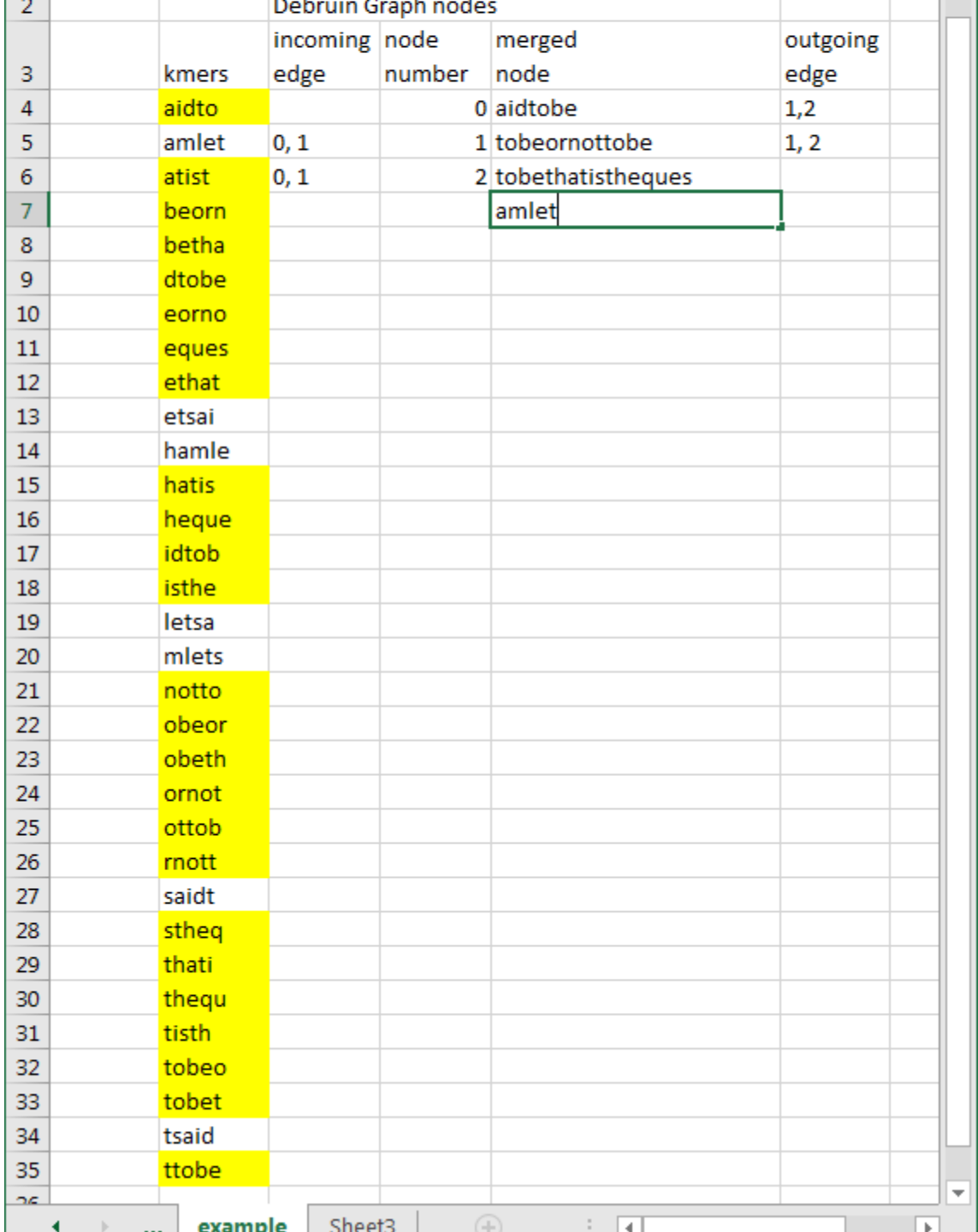
De Bruijn Graph text reconstruction

- I continued tracing node 1 (tobeo) until I reach a branch.
- Since both words beginning tobe are yellow, I know I have already used these words. They turn out to be the beginning of nodes 1 and 2. This is a repeated phrase that forms a cycle in the graph.
- The branch connects the right edge of node 1 to the beginning of nodes 1 and 2.
- Update the outgoing edge of node 1, and the incoming edges of both node 1 and 2.
- Continue tracing node 2.

2		Debruin Graph nodes			
3	kmers	incoming edge	node number	merged node	outgoing edge
4	aidto		0	aidtobe	1,2
5	amlet	0, 1	1	tobeornottobe	1, 2
6	atist	0, 1	2	tobet	
7	beorn				
8	betha				
9	dtobe				
10	eorno				
11	eques				
12	ethat				
13	etsai				
14	hamle				
15	hatis				
16	heque				
17	idtob				
18	isthe				
19	letsa				
20	mlets				
21	notto				
22	obeor				
23	obeth				
24	ornot				
25	ottob				
26	rnott				
27	saidt				
28	stheq				
29	thati				
30	thequ				
31	tisth				
32	tobeo				
33	tobet				
34	tsaid				
35	ttobe				

De Bruijn Graph text reconstruction

- At this point, I've run out of overlapping words for node 2, since there is no word beginning ques. All current existing nodes (1,2,3) are unable to be extended.
- Since I haven't used up all the words, I choose one at random and begin a new node (3).
- I choose amlet.



Debruin Graph nodes

	kmers	incoming edge	node number	merged node	outgoing edge
4	aidto		0	aidtobe	1,2
5	amlet	0, 1	1	tobeornottobe	1, 2
6	atist	0, 1	2	tobethatistheques	
7	beorn			amlet	
8	betha				
9	dtobe				
10	eorno				
11	eques				
12	ethat				
13	etsai				
14	hamle				
15	hatis				
16	heque				
17	idtob				
18	isthe				
19	letsa				
20	mlets				
21	notto				
22	obeor				
23	obeth				
24	ornot				
25	ottob				
26	rnott				
27	saidt				
28	stheq				
29	thati				
30	thequ				
31	tisth				
32	tobeo				
33	tobet				
34	tsaid				
35	ttobe				

example Sheet3

De Bruijn Graph text reconstruction

- I Continue extending node 3 until I reach a used word (aidt). This tells me that node 3 connects to the node beginning aidt (node 1)
- I mark that node 3 has an outgoing connection to node 1, and that node 1 has an incoming connection with node 3
- Node 3 is now complete. There is only one word left (hamle) which I can tell by inspection is at the left end of node 3

2		Debruin Graph nodes			
3	kmers	incoming edge	node number	merged node	outgoing edge
4	aidto		0	aidtobe	1,2
5	amlet	0, 1	1	tobeornottobe	1, 2
6	atist	0, 1	2	tobethatistheques	
7	beorn		3	amletsaidt	0
8	betha				
9	dtobe				
10	eorno				
11	eques				
12	ethat				
13	etsai				
14	hamle				
15	hatis				
16	heque				
17	idtob				
18	isthe				
19	letsa				
20	mlets				
21	notto				
22	obeor				
23	obeth				
24	ornot				
25	ottob				
26	rnott				
27	saidt				
28	stheq				
29	thati				
30	thequ				
31	tisth				
32	tobeo				
33	tobet				
34	tsaid				
35	ttobe				

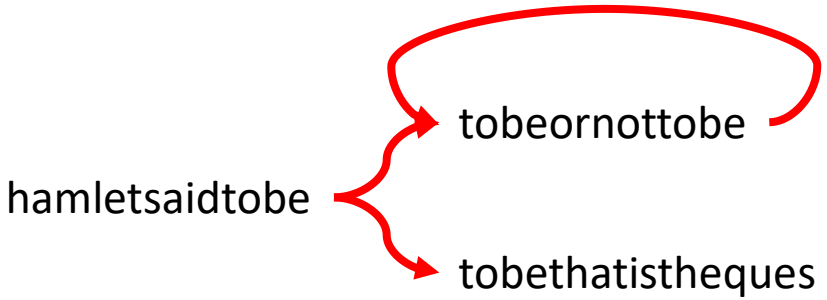
De Bruijn Graph text reconstruction

- Just a little cleanup, and I'm done.
- Because we started with a random word, some of our nodes may begin in the middle of a branchless stretch
- Node pairs where the left node has only one outgoing edge, and the right has only one incoming edge are connected without branches and can be merged
- Node 3 (left) and node 0 (right) are such edges

2			Debruin Graph nodes			
3		kmers	incoming edge	node number	merged node	outgoing edge
4		aidto	3	0	aidtobe	1,2
5		amlet	0, 1	1	tobeornottobe	1, 2
6		atist	0, 1	2	tobethatistheques	
7		beorn		3	hamletsaidt	0
8		betha				
9		dtobe				
10		eorno				
11		eques		0	hamletsaidtobe	1,2
12		ethat	0, 1	1	tobeornottobe	1, 2
13		etsai	0, 1	2	tobethatistheques	
14		hamle				
15		hatis				
16		heque				
17		idtob				
18		isthe				
19		letsa				
20		mlets				
21		notto				
22		obeor				
23		obeth				
24		ornot				
25		ottob				
26		rnott				
27		saidt				
28		stheq				
29		thati				
30		thequ				
31		tisth				
32		tobeo				
33		tobet				
34		tsaid				
35		ttobe				

De Bruijn Graph text reconstruction

- Final graph
- Interpretation
- Hamlet said “To be, or not to be, that is the ques…”



De Bruijn Graph text reconstruction

De Bruijn Graph text reconstruction

De Bruijn Graph text reconstruction

De Bruijn Graph text reconstruction