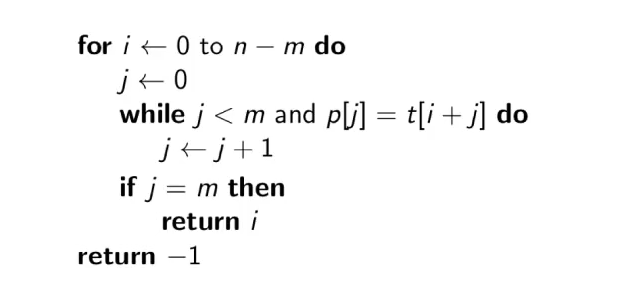
**HOW DOES BRUTE FORCE STRING MATCHING ALGORITHM WORK?**

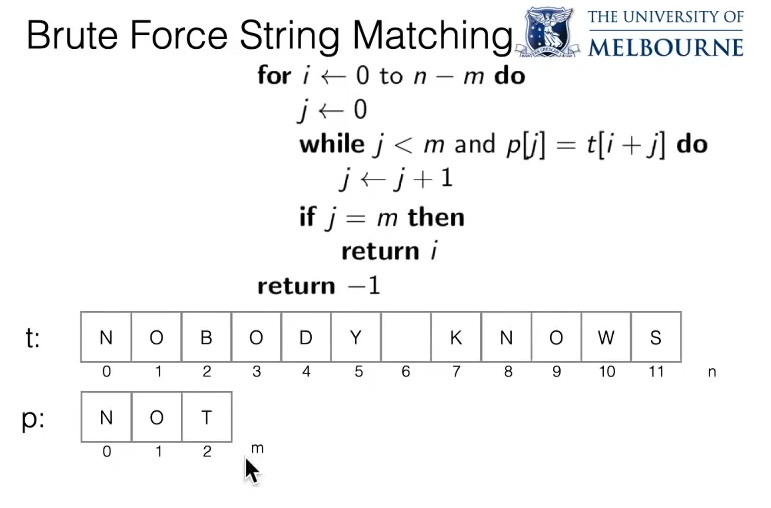
Pattern P: A string of m characters to search for.

Text t: A string of n characters to search in or an array of characters that we are searching in.

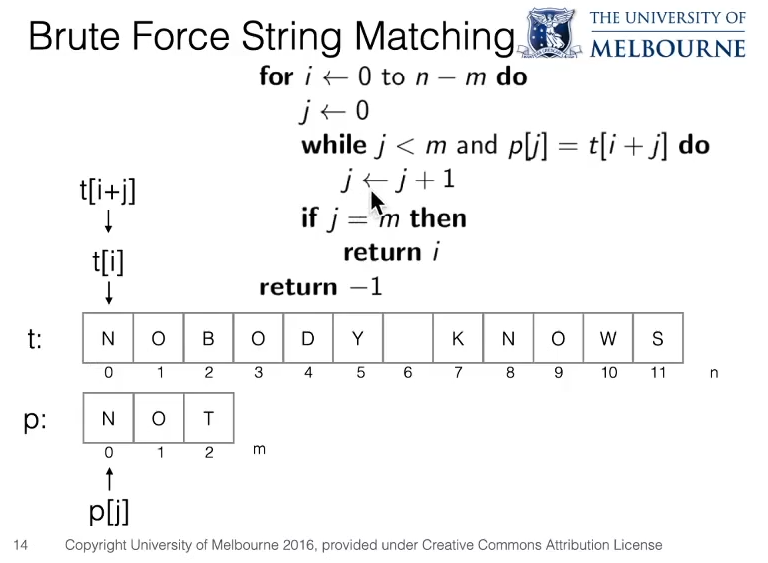
* The pointer i is used to run through the text and the pointer j is used to run through the pattern.



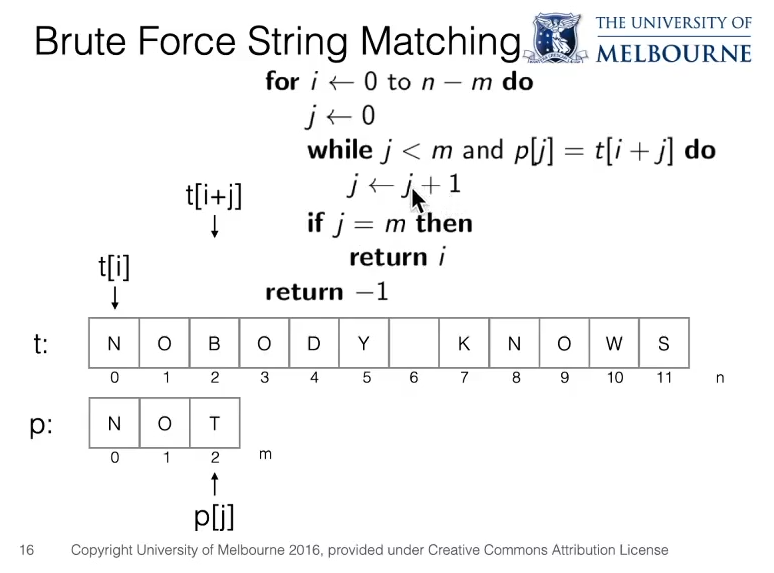
We would assign the text t as n characters long and the pattern as m characters long.



0 to n-m every position where the pattern would begin.



* It’s actually the value of j that’s incrementing in the algorithm. When the comparison reaches the end of the pattern then the value of j is reset and the value of i gets incremented.



* In the above algorithm, i is held static.
* t[i+j] would correspond to the pointer in the text and the p[j] would correspond to the pointer in the pattern.
* We do the comparison from left to right.
* Initially N(In text t) is equal to N(In pattern p) so , we do j = j+1 as  went to be true.
* So we do j = j+1 , now p[j] = 0 and j = 1 . t[i+j] would be equal to 0+1 = 1. And then O at t[i+j] or t[0+1] would match p[j] or p[1] equal to O, as they would match each other we do j = j+1. The condition  came out to be TRUE.
* Next j = 2 so i+j would be 0+2 = 2 so, p[j] = T and t[i+j] would be equal to B. So the condition  is not TRUE, as p[j] is not equal to t[i+j].
*  j would be equal to m when we would have a perfect match with the text T so that all the characters would match and j would be incremented m times or say for example in the , in the given pattern array NOT , say if the text would also contain NOT, the value of j would be incremented 3 times, such that j would have a value of 3 or m.

Text (T):

|  |  |  |  |
| --- | --- | --- | --- |
| N | O | T | T |

0 1 2 3

Initially J is 0, N and N would match so j would become 1, O and O would match so j becomes and then T and T would match so j becomes 3. So we have the value of j equal to 3 or m. When j equals m we would return i .

|  |  |  |
| --- | --- | --- |
| N | O | T |

0 1 2

Pattern (P)