

# Data Structures and Algorithm COMP9024 25T0

## T13A & T14A lab

January 16<sup>th</sup> 2025 Thursday

### **Content of today:**

1. singly linked list & doubly inked list
2. Assessment 1
3. Q & A

Joffrey Ji ( [z5450981@ad.unsw.edu.au](mailto:z5450981@ad.unsw.edu.au) )



## Question 2\_1\_1

(Counting primitive operations)

The following algorithm

- takes a sorted array  $A[1..n]$  of characters
- and outputs, in reverse order, all 2-letter words  $v\omega$  such that  $v \leq \omega$ .

```
→ for all  $i=n$  down to 1 do  
→   for all  $j=n$  down to  $i$  do  
     print " $A[i]A[j]$ "  
   end for  
end for
```

Count the number of primitive operations (evaluating an expression, indexing into an array). What is the time complexity of this algorithm in big-Oh notation?

for (int i = n ; i >= 1 ; i--)

n = 3

i >	i = 3	i ≥ 1 ✓	i --
i >	i = 2	i ≥ 1 ✓	i --
i >	i = 1	i ≥ 1 ✓	i --
i >	i = 0	i ≥ 1 ✗	

$$3 + 3 + 1 \Rightarrow \underline{1 + n + 1}$$

for (int j = n ; j >= i ; j--)

i = n = 3

print(1)

i > j = 3

j >= i, j--

i > j = 2

j >= i (X) total = 3

i = 2

print(2)

i > j = 3

j >= 2, j--

i > j = 2

j >= 2, j--

$$| i > j = 1 | | j \geq 2 \text{ (X) total} = 5$$

$$i = 1$$

print(3)

$$i > j = 3$$

$$j \geq 1 \quad j \text{ ---}$$

$$i > j = 2$$

$$j \geq 1 \quad j \text{ ---}$$

$$i > j = 1$$

$$j \geq 1 \quad j \text{ ---}$$

$$i > j = 0$$

$$j \geq 1 \text{ (X) total} = 7$$

$$3 + 5 + 7 + 9 + \dots + 2n + 1 = \underline{n(n+2)}$$

for print:  $A[i] \quad 1 + 2 + \dots + \underline{n}$

$$A[j] \quad 1 + 2 + \dots + \underline{n}$$

$$\frac{(1+n)n}{2} \times 2$$

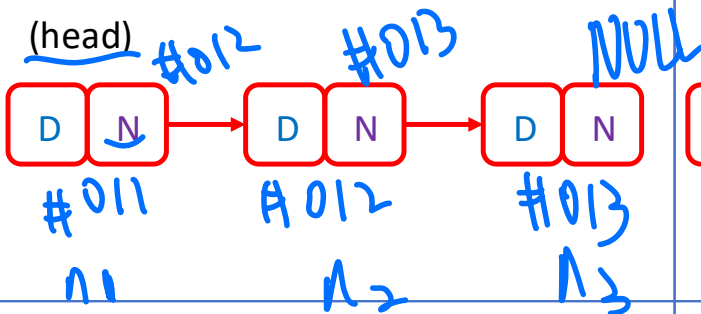
total

$$2n^2 + 5n + 1 \rightarrow \underline{O(n^2)}$$

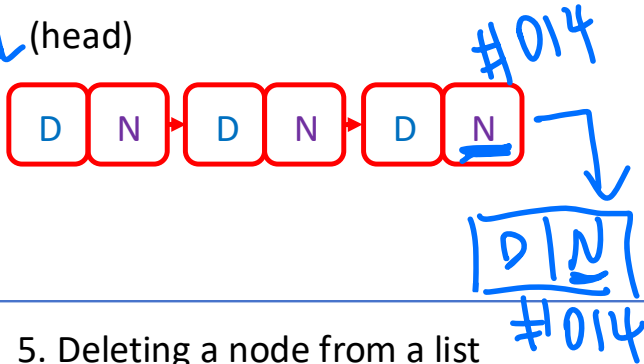


## Operation of Singly Linked List

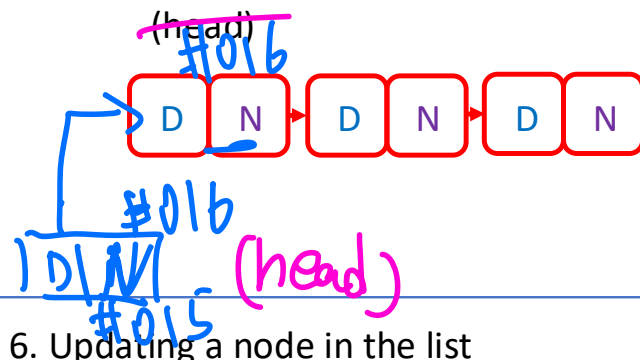
### 1. Traversing a singly linked list



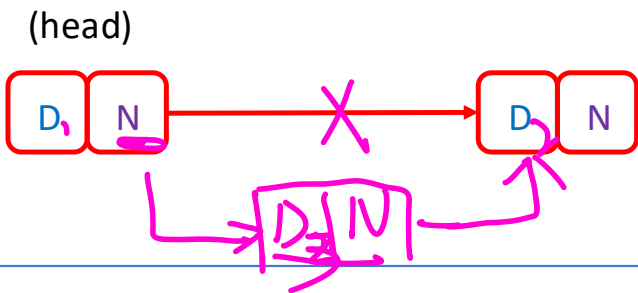
### 2. Append a node to the end of a list



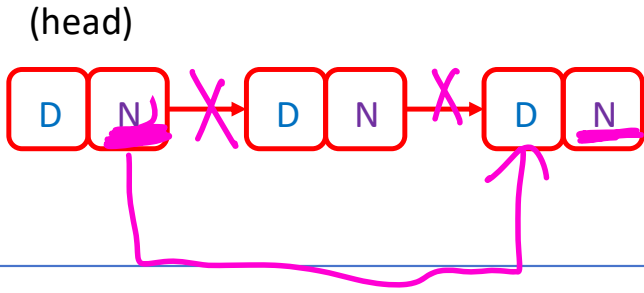
### 3. Prepend a node to the start of a list



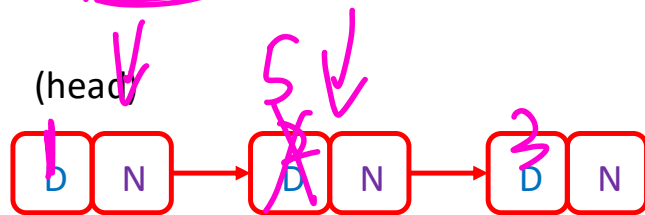
### 4. Inserting a new node to a specific position of a list



### 5. Deleting a node from a list



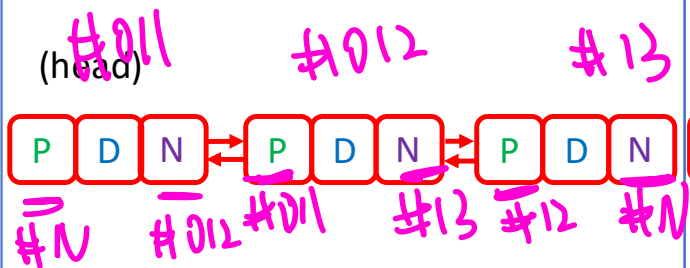
### 6. Updating a node in the list



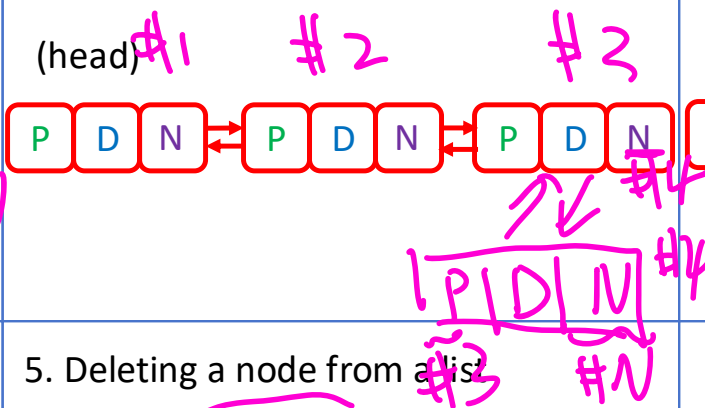


## Operation of Doubly Linked List

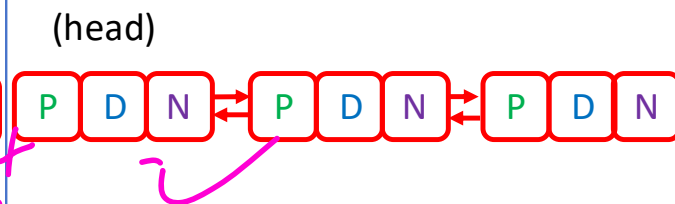
1. Traversing a doubly linked list



2. Append a node to the end of a list



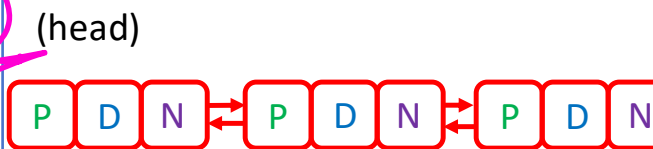
3. Prepend a node to the start of a list



4. Inserting a new node to a specific position of a list



5. Deleting a node from a list



6. Updating a node in the list



## Question 3

(Algorithms and complexity)

Develop an algorithm to determine if a character array of length contains at least one letter more than once. For example, "repeat" is such a word but "until" is not.

- Write the algorithm in pseudocode.
- Analyse the time complexity of your algorithm.
- Implement your algorithm in C as a function

```
bool hasRepeatedLetter(char A[])
```

that returns `true` if a letter is repeated in string `A`, and `false` otherwise.

*Hint:* The standard library `<stdbool.h>` defines the basic data type `bool` with the two values `true` (internally encoded as 1) and `false` (= 0).

- Use your solution to Exercise c. to write a program that prompts the user to input a word and checks whether it has a repeated letter. Examples of the program executing are

```
prompt$ ./word
Enter a word: repeat
yes
prompt$ ./word
Enter a string: until
no
```

*Hints:*

- You may assume that the input consists of lower case letters only.
- You may assume that the input consists of no more than 31 characters (excluding the terminating `'\0'`).
- You can use the standard I/O library function `scanf("%31s", str)` to read a word from the input.
- You may use the standard library function `strlen(char[])`, defined in `<string.h>`, which computes the length of a string (without counting its terminating `'\0'`-character).

We have created a script that can automatically test your program. To run this test you can execute the `dryrun` program that corresponds to this exercise. It expects to find a program named `word.c` in the current directory. You can use `dryrun` as follows:

```
prompt$ 9024 dryrun word
```





# Data Structures and Algorithm COMP9024 25T0

## T13A & T14A lab

January 16<sup>th</sup> 2025 Thursday

**~Questions and Suggestions~**

Thanks for listening

Joffrey Ji ( [z5450981@ad.unsw.edu.au](mailto:z5450981@ad.unsw.edu.au) )