1 Fundamentals

Higher Grade Problem

Time complexity:

If N elements in list - compare N times
If match is found - dequeue element

first case: if match is found for last element

In first case program has to iterate through N elements (O(N)) and call (+1) dequeue() to iterate through N elements (räknas ej) again to find last element to delete (O(N))
→ O(2N)

second case: if match is found for every element

In second case program has to iterate through N elements (O(N)) and then call (+N) dequeue() for every N. dequeue() will always be called for the first element in list, meaning head has to be redirected N times.
→ O(3N)

Worst case: O(3N) = O(N)

1. Fundamentals

Higher Grade Problem

Memory Complexity:

If N elements in list → N memory spaces

+ constant memory occupants

So...

- N elements
- 1 comparable element
- 1 head node
- 1 counter
- 1 size

Which gives:

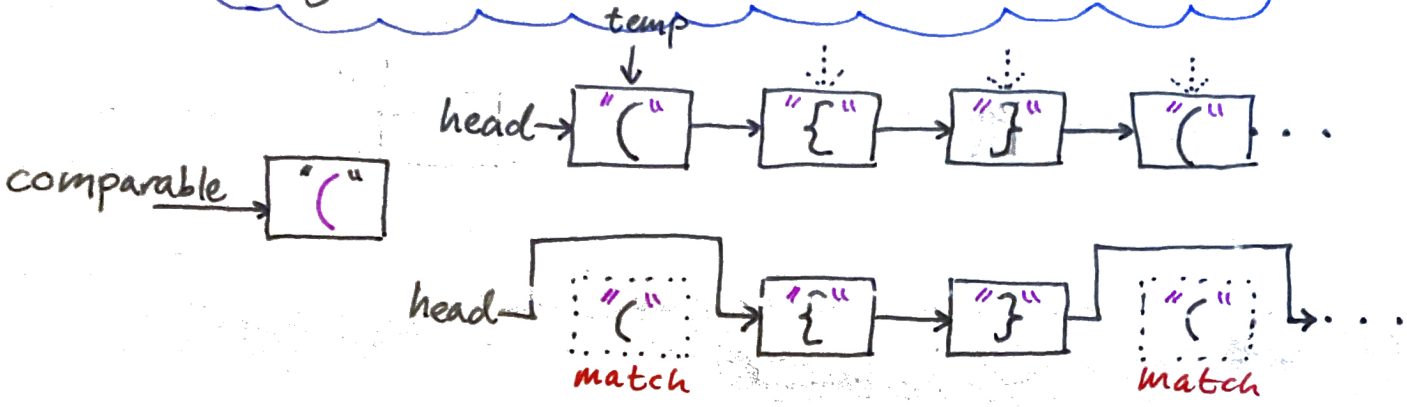
N + 4, but as $N \to \infty$ consants becomes insignificant

$\longrightarrow O(N) = \Omega(N)$

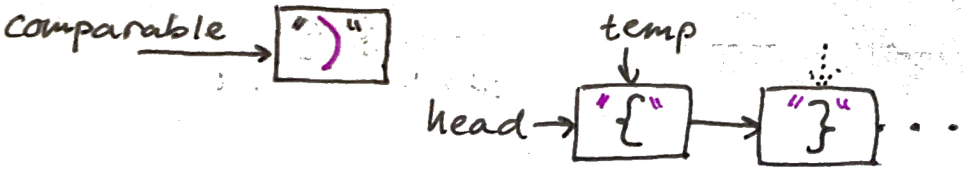
Worst case = every case

1. Fundamentals

Higher Grade Problem

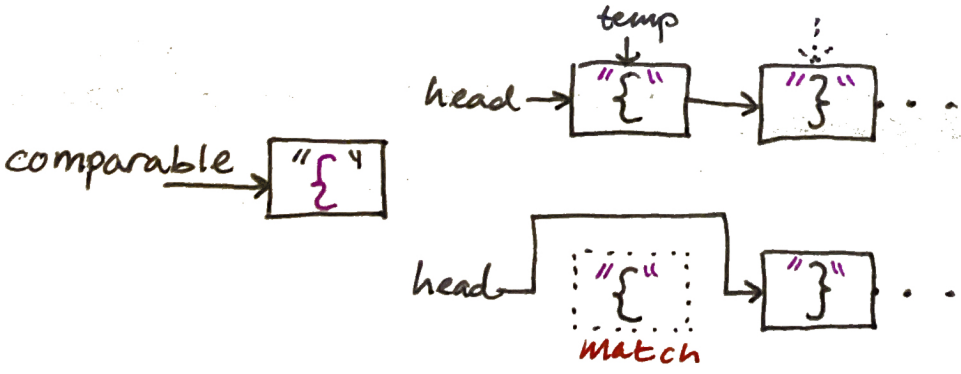


count = 2

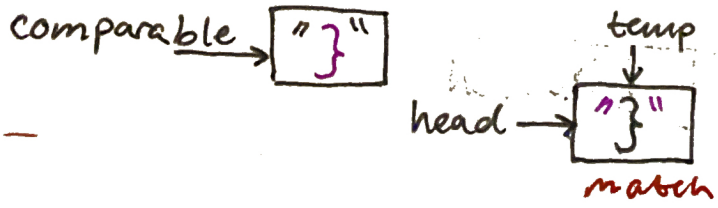


count = 0

( ) is not balanced



count = 1



count = 1

{ } is balanced