

## Text justification

For this problem we need to format lines so that each line is fully justified to max width characters.

### Key Rules:

1. Greedy packing: Put as many words as possible in a line without exceeding max width.
2. Fully justify (middle lines):
  - Spaces distributed evenly between words.
  - If spaces don't divide evenly, leftmost gaps get more spaces.
3. Text line or single word line: Left-justified: words separated by single spaces, and pad the rest at the end.

### Algorithm:

1. Initialize pointers to collect words per line:
  - Count words until adding another word would exceed max width.
2. Compute total spaces:
  - $\text{spaces} = \text{maxWidth} - \text{sum}(\text{length}(\text{words}))$
3. For middle lines:
  - $\text{slots} = \text{words} - \text{count} - 1$
  - Distribute spaces evenly:
    - $\text{space\_each} = \text{spaces} / \text{slots}$
    - $\text{extra} = \text{spaces} \% \text{slots}$  (add 1 extra space to first extra gaps)
4. For last line:
  - Join words with single space and pad right with spaces.

### Complexity:

- Time:  $O(N)$  where  $N$  = total number of characters (single pass).
- Space:  $O(N)$  for result storage.