String/sequence matching and search

- Prefix search:
  - Find all strings that start with “tab”:
    - “table”; “tabular”; “tablet”; ….

- Subsequence search:
  - Find all strings that contain the subsequence “ark”:
    - “marketing”; “spark”; “quark”
  - Find all occurrences of “acd”:
    - “aabacdcdabdcababdacddcab.”

- Sequence similarity:
  - “table” vs. “cable”?
  - “table” vs. “tale”?
  - “table” vs. “tackle”?

Subsequence/Pattern Search

data: abcbbbaabbaabcbbbaaabbccbbbaabbaacbbbaabbccbcbbbaabcbbbaabab

pattern: abbcc

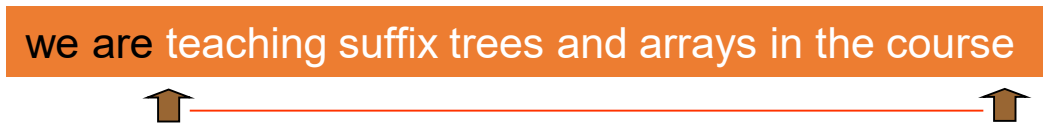
- Brute force approach:
  - scan the sequence, while aligning the pattern for each position in the sequence
- Given a sequence of length N, and pattern of length M
  - Cost: O(NxM)
  - For the above example, cost: 60 x 5

Suffix Trees and Arrays

- Tries work well if we search for a prefix

- Suffix trees and suffix arrays
  - Input text: a single long string
  - each position in the text gives a suffix

  - alternatively, start of each word in the text gives a suffix



Suffix Trees

- Suffix trees
  - Input text: a single long string
  - each word position in the text gives a suffix