Data Structures with C++ : CS189

Lecture 15-1: String Matching

Recap: This Semester

- We've looked inside some data structures and seen how they work
- We've debugged, optimized, and refactored actual code projects
- We sobbed through recursion
- We've looked at some of the top Graph Theory algorithms
- Today is one of top examples of "Interview Algorithm"
  - You will only ever see inside this algo today, and your job interview

String Searching

- Ctrl + F inside a word doc is only the most obvious reason you'd want to find a string inside a string
- If you turn "string" in that sentence in to "pattern" you can see other uses for this
  - There are only four characters in DNA. GTCCCGCTGGCG is the gene for hating cilantro. Do you have it?

"How do I find a string inside another?"

- Brute Force:
  - Scan Text letter by letter
  - If a letter matches with the first letter of the Pattern, then loop through the pattern to see if the whole thing is matched
  - If not, go back to looping through the text.
- Essentially O(n^2) since every letter of P might check against every letter of T
  - Math for actual big o is long and annoying
    - The longer the search pattern the fewer times you need to check, etc
- Hancart says start at the second letter to get through faster since you can jump forward two at a time.
  - First letter of pattern is checked last

Knuth - Morris - Pratt

- Brute said to shift 1 on a miss
- Hancart says to shift 1 or 2 on a miss
- What we really want is to shift as far as we can based on what matched and what didn't
  - Search for ababab in ababx
  - First two match
  - Third realises that it might be the start of a match
    - Think of a "hit inside a miss"
  - When the pattern fails, jump up to that bookmark
    - Plus one - the presence of that bookmark proves the first is a match

KMP: Speed up Table

P = ababaxxx
Where did the pattern match fail?
0? is -1 - brute base case
1? - take ab and try to overlap ab. 0 - 1 = 0
2? aba? 1 - 1 = 0
3? abab, so 2 - 1 = 1
4? ababa is 3-1. aba is front and back

Result -1 0 0 1 2 0 0 0

- So how far should we skip in case a hit started inside the miss?
- Make a table of "If the Pattern fails at this index, move this far forward"
  - Brute's answer is always 1
  - KMP's answer is "How far can the Pattern *at this point* become its own suffix?"
    - The end of last check is the beginning of this
    - Whole pattern doesn't count since that's not what suffix means
  - Easy to precompute since it's independent of T
    - For algo reasons, the number is "extra" letters so they each are -1, capped at 0

Boyer - Moore

p685 has a table for this one

- KMP started at the start of P and T and decided how many spots to skip
- BM starts at the start of T still, but the end of P.
  - When you find a conflict as you loop through P, you don't need to keep searching left - P can't fit between the conflict point and T's current spot
- So pull P to the right until you pass the conflict spot
  - If P has the desired first letter left of the conflict spot, stop early (The KMP table effectively)
  - Sorta pulls from the right where KMP pushed from the left

Suffix Array

p. 717

- Suffix = all partial words from the right
  - suffix, uffix, ffix, fix, ix, x
    - "Proper" suffix skips full word (KMP table)
  - We don't store *prefixes* since you don't know if they are finished or not
- Take every word and put it and all its suffixes in one big array
  - Yes, this is a slow pre-process sort of step
- Alphabetize that array
  - Just store a number for where that pattern starts instead of tons of extra words
- Binary search can now find any pattern
  - Prefix, whole word, and suffix are normal searches. Mid-fix is just the prefix of a suffix

Suffix Array ++

- Longest Common Prefix (LCP) is used in some advanced algos beyond this class
  - It's an extra number stored by the StartPosition that has "How many letters do I share from the start with the entry before me?"
- Can also refer to this process as Suffix Tree if you store the data differently.
  - Tree-style is faster at run time, but bigger and fragmented
  - Array-style allocates one array at the start and that's it
    - Can even be appended to a file itself

End