CPCFI

Lecture: STRING MATCHING II

Unit: 9 - STRING PROCESSING + COMPUTATIONAL GEOMETRY

Instructor: WESTON

9.1.2. STRING MATCHING ON A 2D GRID

↳ GIVEN A 2D GRID FIND THE OCURRENCE(S) OF PATTERN P IN THAT GRID.

→ SEARCH CAN BE MADE IN 4 OR 8 DIRECTION
→ PATTERN CAN BE FOUND IN A STRAIGHT LINE OR DIAGONAL

```
A B C D E F G H I G G
H E B E W A L D O R K
F T Y A W A L D O R M
F T S I M R L Q S R C
B Y O A R B E D E Y V
K L L B Q W I K O M K
S T R E 8 G A D H R B
Y U I Q L X C N B J F
```

* SOLUTION CAN BE THOUGHT AS GRAPH SEARCHING
* ALSO DEPTH LIMITED SEARCH WHERE THE DEPTH IS THE LENGTH OF THE PATTERN

EXTRA: Z-FUNCTION

↳ FOR A GIVEN STRING S, THE Z-FUNCTION IS AN ARRAY OF n LENGTH WHERE THE i-TH ELEMENT IS EQUAL TO THE GREATEST NUMBER OF CHARS. STARTING FROM POSITION i THAT COINCIDE WITH THE FIRST CHARS. OF S

- Z[i] : LENGTH OF THE LONGEST STRING THAT IS A PREFIX OF S AND A PREFIX OF THE SUFFIX OF S STARTING AT i

EXAMPLE:

"AAAAA"	"AACBACCA"	"ABACABA"	"ABCDE"
Z=[0 4 3 2 1]	Z=[0 1 0 0 1 0 0 1]	Z=[0 0 1 0 3 0 1]	Z=[0 0 0 0 0]

* GO TO TRIVIAL IMPLEMENTATION

↳ $O(n^2)$

EFFICIENT ALGORITHM : $O(n)$

IDEA → COMPUTE $Z[i]$ FROM $1, \dots, n-1$ (ZERO BASED) TRYING TO USE PREVIOUSLY COMPUTED VALUES

↳ WE'LL KEEP THE $[L, R]$ INDICES OF THE RIGHTMOST SEGMENT MATCH

SUBSTRINGS THAT COINCIDE WITH A PREFIX OF S

- L: STARTING INDEX OF S.B.
- R: ENDING " " " → BOUNDARY OF WHERE WE HAVE SCANNED STRING S

$\boxed{\text{S.M.:}}$

$S_0 S_1 S_2 \dots S_i S_{i+1} \dots S_{n-1}$

SEGMENT MATCH

$S_0 S_1 = S_i S_{i+1}$

↳ LENGTH OF S.M. : $\boxed{i + Z[i] - 1}$

ALGORITHM :

TWO CASES :

1. $i > R$: CURRENT POSITION IS OUTSIDE OF SEGMENT MATCH

   - COMPUTE $Z[i]$ USING TRIVIAL ALGORITHM
   - IF $Z[i] > 0$ WE UPDATE VALUES L, R SINCE LENGTH OF S.M. IS BETTER THAN PREVIOUS R

2. $i \leq R$: CURRENT POSITION IS INSIDE S.M. $[L, R]$

   - INITIALIZE $Z[i]$ TO SOME VALUE BETTER THAN ZERO

OBSERVATION :

- $S[L \dots R] = S[0 \dots R-L]$ THESE SUBSTRINGS MATCH

$$\underbrace{S_0 S_1 S_2}_{R-L} \dots \underbrace{S_L S_{L+1} \dots S_R}_{R-L} \dots S_{n-1}$$

- FOR INITIAL VALUE $Z[i] = Z[i-L]$

↑

VALUE FOR SEGMENT MATCH $S[0 \dots R-L]$

- Z[i - L] MIGHT BE TOO LARGE FOR CURRENT i INDEX THUS :

$$Z[i] = \min(\underline{R - i + 1},\ Z[i - L])$$

IF WE ARE AT INDEX
$i = R = n - 1$ THEN MAX
POSSIBLE VALUE WITHOUT
EXCEEDING INDEX R IS 1

- AFTER INITIALIZING Z[i] WE TRY TO INCREMENT Z[i] USING THE TRIVIAL ALGORITHM

* GO TO Z_FUNCTION.CPP

APPLICATIONS :

- [SEARCH THE SUBSTRING]

  - FIND OCCURRENCES OF STRING s IN TEXT t
  - CONCATENATE $s\#t$
  - BUILD Z-FUNCTION
  - IF $k = |s|$ → OCCURRENCE OF s IN t AT INDEX i : $i \in [0, \dots, |t|-1]$

    ↳ $k = z[i + |s| + 1]$

- [NUMBER OF DISTINCT SUBSTRINGS IN A STRING] $O(n^2)$

  - COUNT # OF DIFFERENT SUBSTRINGS OF STRING s $\quad n = |s|$
  - APPEND NEW CHAR c TO s
  - DEFINE STRING $T = c + s$ (CONCATENATION)
  - REVERSE T
  - COMPUTE Z FUNCTION OF T
  - GET MAX ELEMENT IN Z → Z_{max}
  - ADD $|T| - Z_{max}$ TO k

    ↳ k INITIALLY AS ZERO

    ↳ STORES # OF DISTINCT SUBSTRINGS

EXAMPLE ↓

ABC $\rightarrow \left| \{A, B, C, AB, AC, ABC\} \right| = k = 6$

ABCA	ABCAB	ABCABC
ACBA	BACBA	CBACBA
$z = [0001]$	$z = [00021]$	$z = [000321]$
$z_{max} = 1$	$z_{max} = 2$	$z_{max} = 3$
$4 - 1 = 3$	$4 - 2 = 2$	$4 - 3 = 1$

$\underbrace{\qquad\qquad}_{k=6}$