LZSS:

* LZSS is an improved variant of LZ77.
* LZ77 → Always outputs a triplet (offset, length, next symbol) for every encoding step.
* LZSS → Uses a flag bit to distinguish between:
  * References to previous data (match)
  * Literal characters (no match).

It doesn't need to encode a "next character" for matches.

Example:

Let's encode the string: ~~abcx~~ abracadabracabra

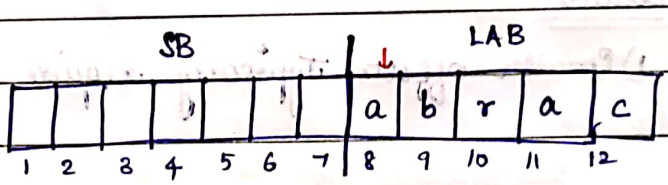
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Search buffer size : 7 characters.

Lookahead buffer size : 5 characters.

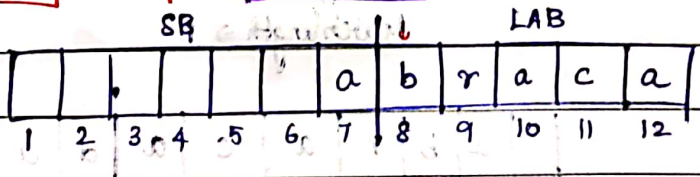
ENCODING:

(i)



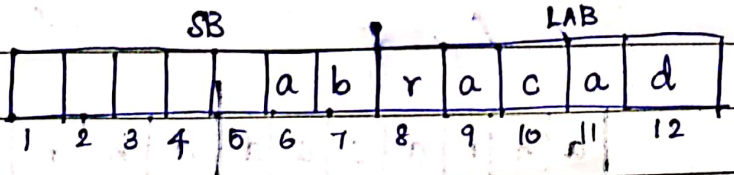
'a': No match Output : <0, a> Move windows

(ii)



'b': No match Output : <0, b> Move window

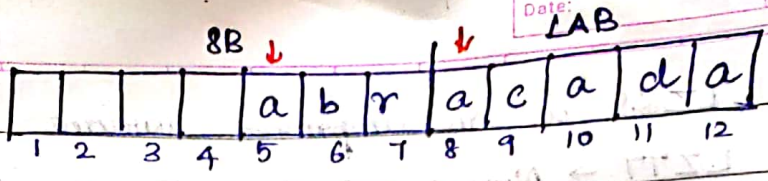
(iii)



'r': No match Output : <0, r> Move window

Scanned with CamScanner

(iv)



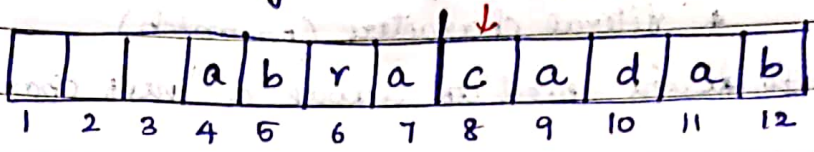
Match found for 'a'.

offset ⇒ 3 Output: 1, <3,1> ← flag indicating match found.

Match length ⇒ 1.

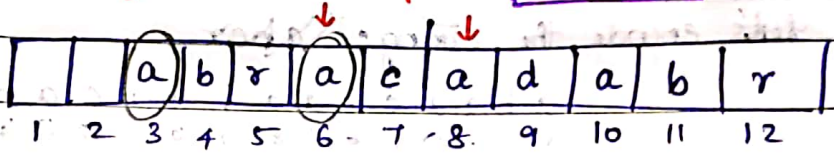
Move window /.

(v)



No match for 'c': Output: <0, c> Move window

(vi)



When multiple ~~character~~ matches of the same length exist, the standard approach is to choose the match with the smallest offset (closest to the current position).

Because:

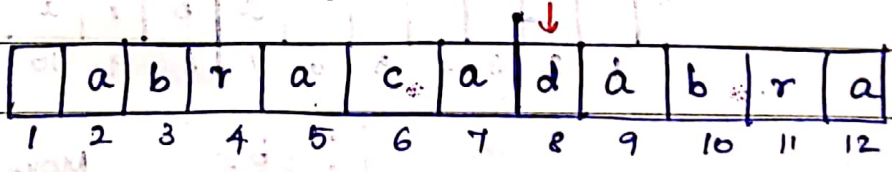
1) Smaller offsets typically require fewer bits to encode.

Match found for 'a'

offset ⇒ 2 Output: 1, <2,1>

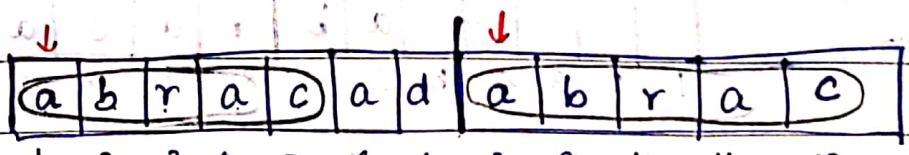
Match length ⇒ 1 Move window.

(vii)



No match for 'd': Output: <0, d> Move window

(viii)



Match found for 'a' which extends to 'abrac'

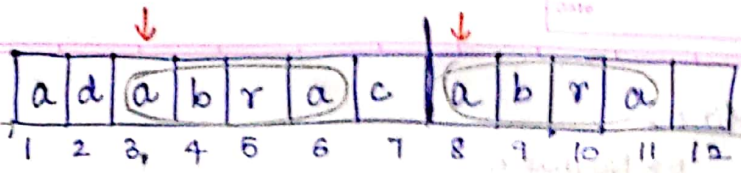
offset ⇒ 7 Output: 1, <7,5>

Match length ⇒ 5

Move window (5 units)

Scanned with CamScanner

(ix)



Match found for 'a' which extends to 'abra'

offset ⇒ 5

Match length ⇒ 4

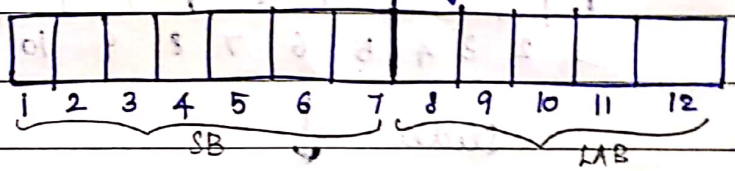
Output ⇒ 1, <5,4>

Final encoded output:

<0,a>; <0,b>; <0,r>; 1,<3,1>; <0,c>; 1,<2,1>; <0,d>; 1,<7,5>; 1,<5,4>

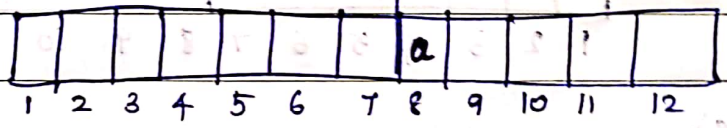
DECODING:

Initial sliding window.

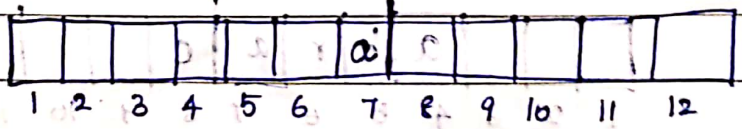


(i) <0,a>.

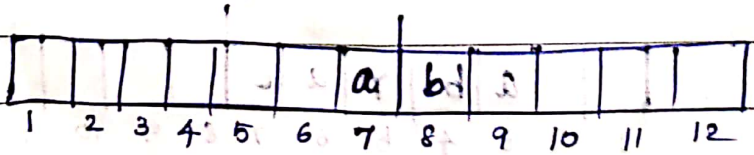
↳ No match:



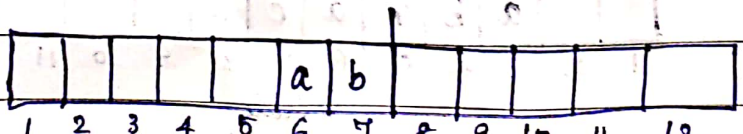
Slide window



(ii) <0,b> ↳ No match.



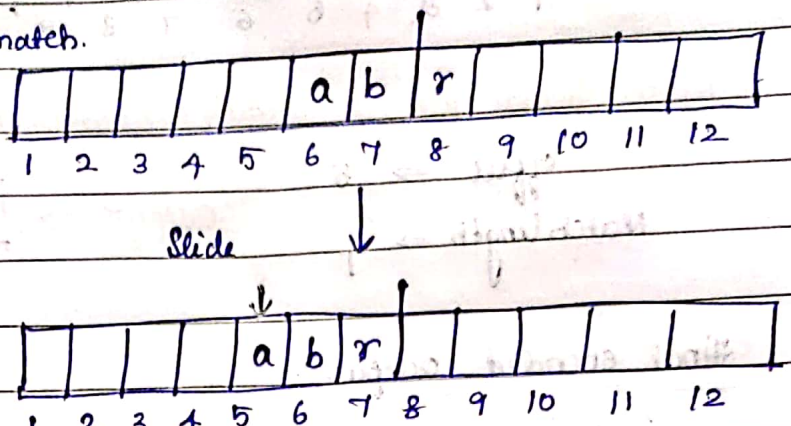
Slide window



Scanned with CamScanner

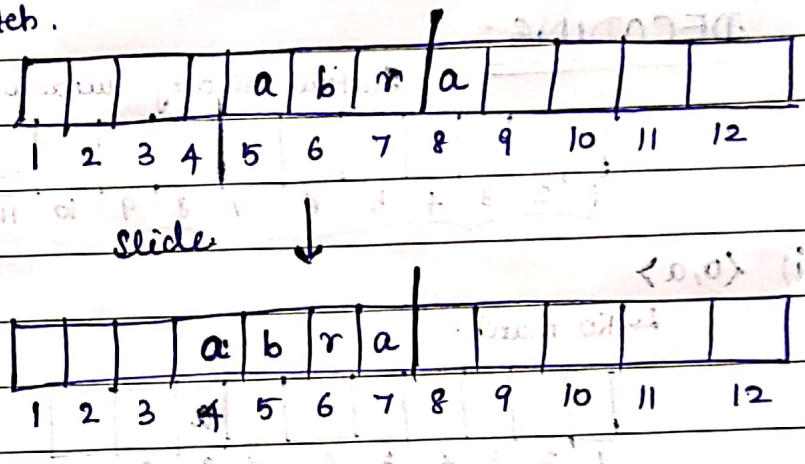
(iii) <0,r>

↳ No match.



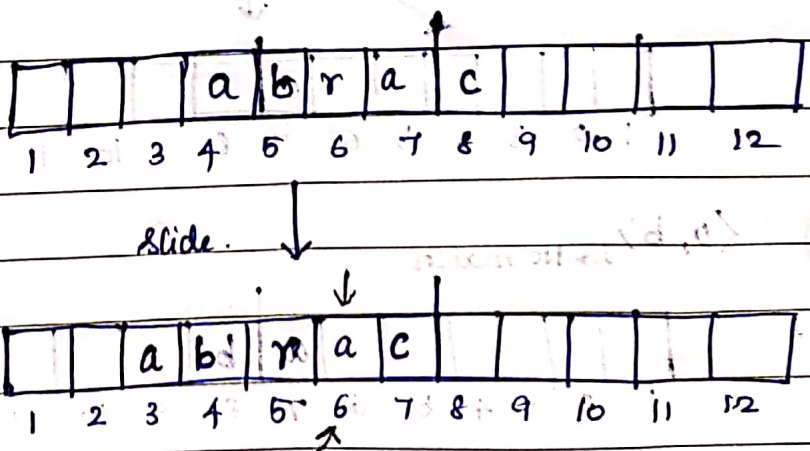
(iv) 1, <3,1>

↳ Match.

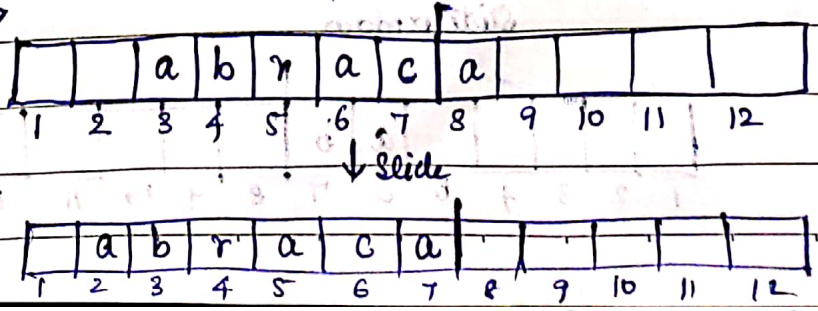


(v) <0, c>

↳ No match.



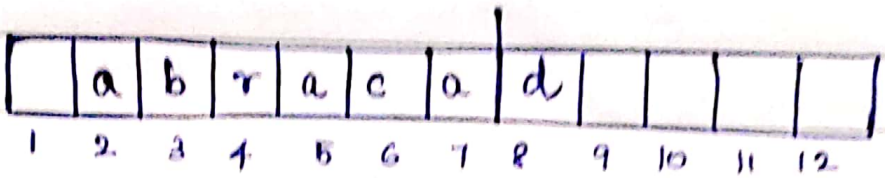
(vi) 1, <2,1>



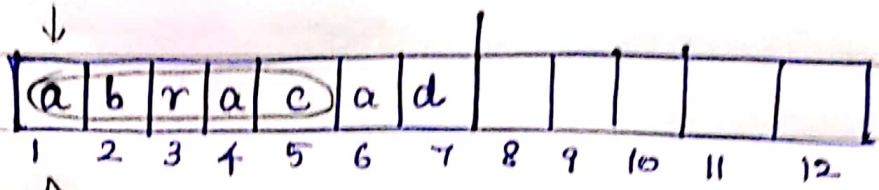
Scanned with CamScanner

(vii) <0, d>

↳ No match.

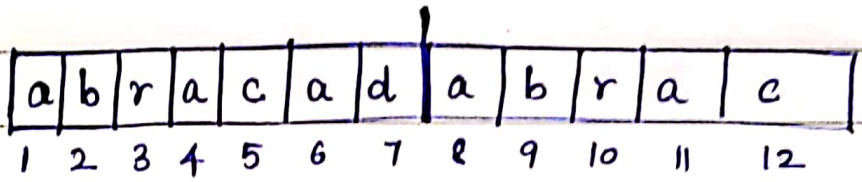


↓ Slide

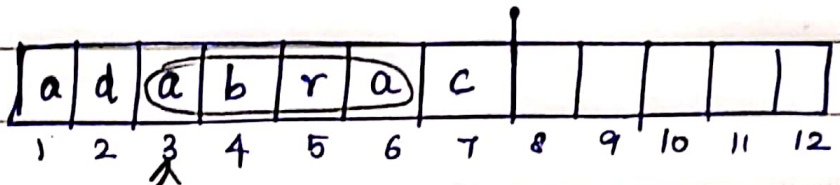


(viii) 1, <7, 5>.

↳ Match.

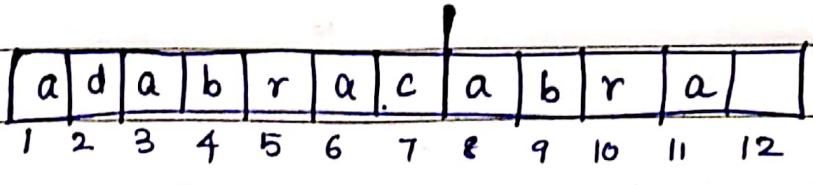


Slide (5 units).



(ix) 1, <5, 4>

↳ Match.



Final decoded sequence: a b r a c a d a b r a c a b r a

Scanned with CamScanner