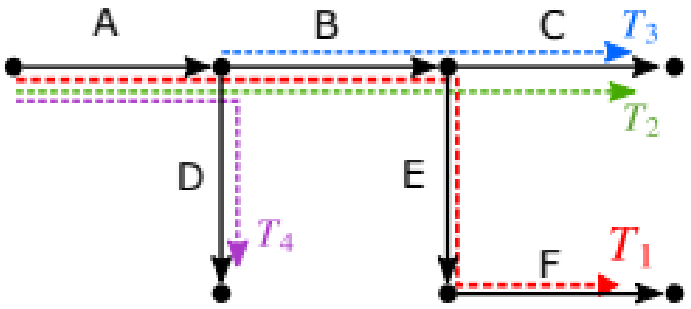
(2018IEEE ICDE)CiNCT Compression and Retrieval for Massive Vehicular Trajectories via Relative Movement Labeling

Definition:



(a) Network-constrained trajectories (NCTs)

$$T = \underbrace{\mathsf{FEBA}}_{T_1^{\mathrm{rev}}} \$ \underbrace{\mathsf{CBA}}_{T_2^{\mathrm{rev}}} \$ \underbrace{\mathsf{CB}}_{T_3^{\mathrm{rev}}} \$ \underbrace{\mathsf{DA}}_{T_4^{\mathrm{rev}}} \$\#. \qquad (1)$$

BWT(Burrows-Wheeler transform)

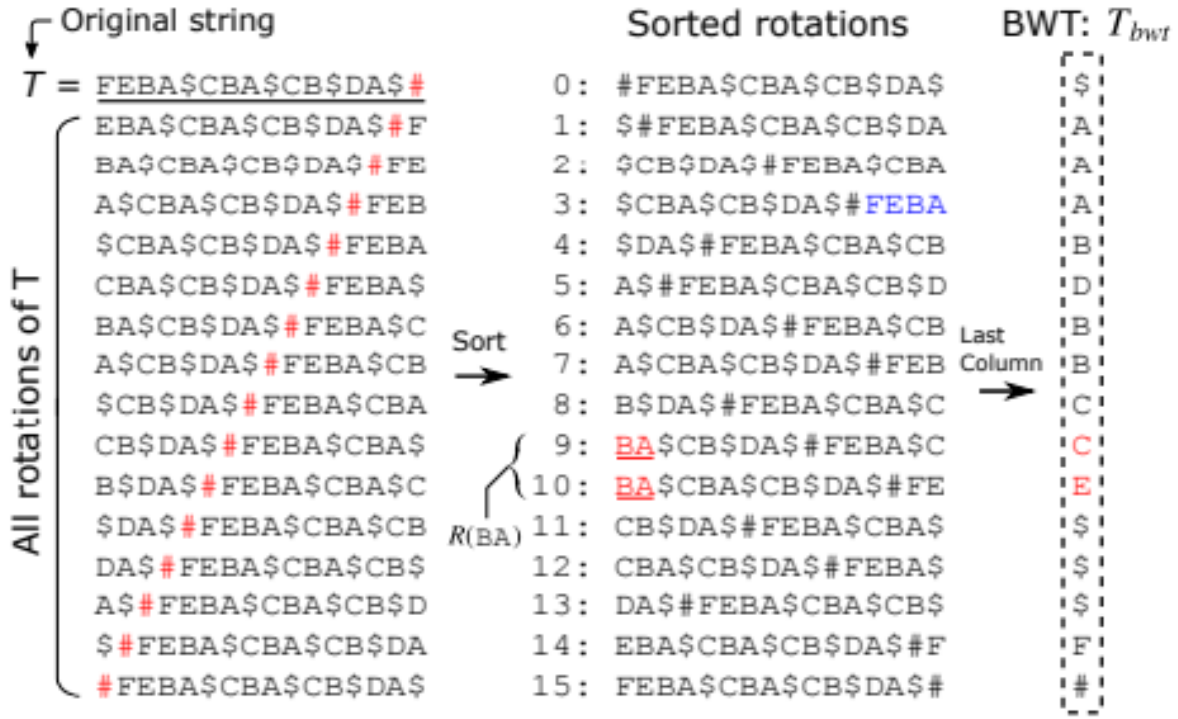


Fig. 2. The BWT of T is defined to be the last column of the sorted rotations of T. This example is based on the trajectory string T in Eq. (1).

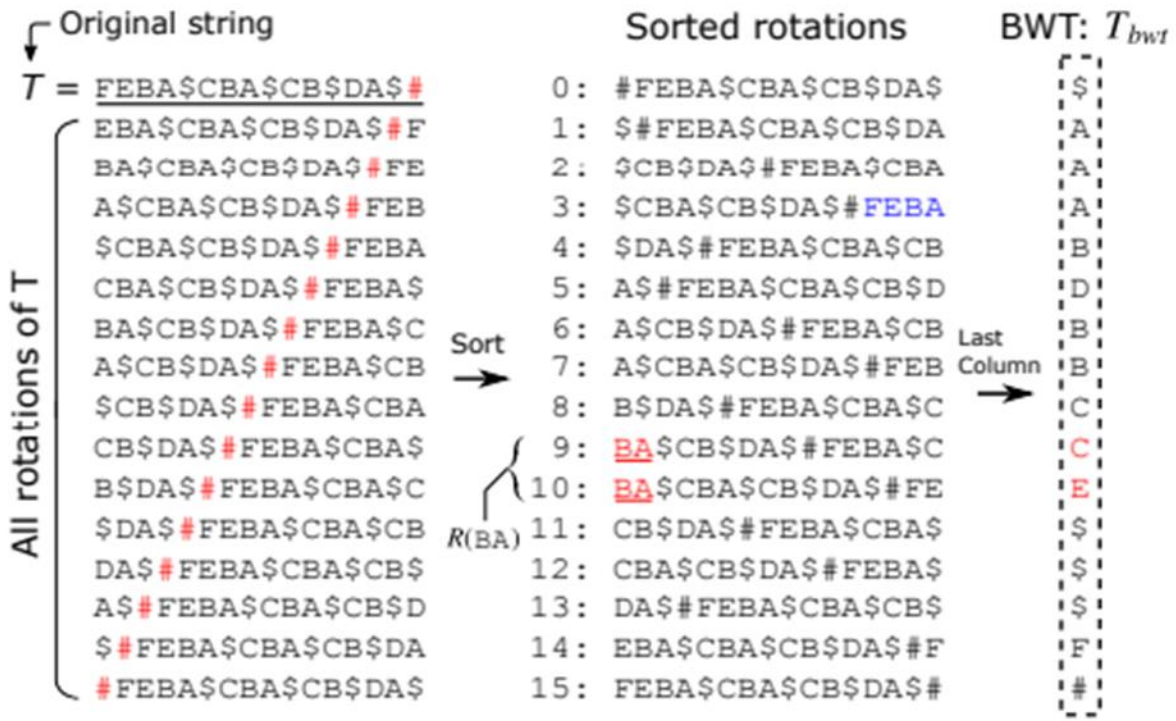


Fig. 2. The BWT of T is defined to be the last column of the sorted rotations of T. This example is based on the trajectory string T in Eq. (1).

$$C[\mathsf{A}] = 5 \text{ and } C[\mathsf{B}] = 8$$

$$rank_{\mathsf{B}}(T_{bwt}, 5) = 1$$

$P = \mathsf{BA}.\quad w = \mathsf{A},\ sp = 5,\ ep = 8.$

Algorithm 1: Finding the suffix range $R(P) = [sp, ep)$ for a given query P of length m based on T_{bwt} (*SearchFM*)

Input: BWT string: T_{bwt}, Query string: P, Integer array: C
Output: Range of T_{bwt} that matches to P

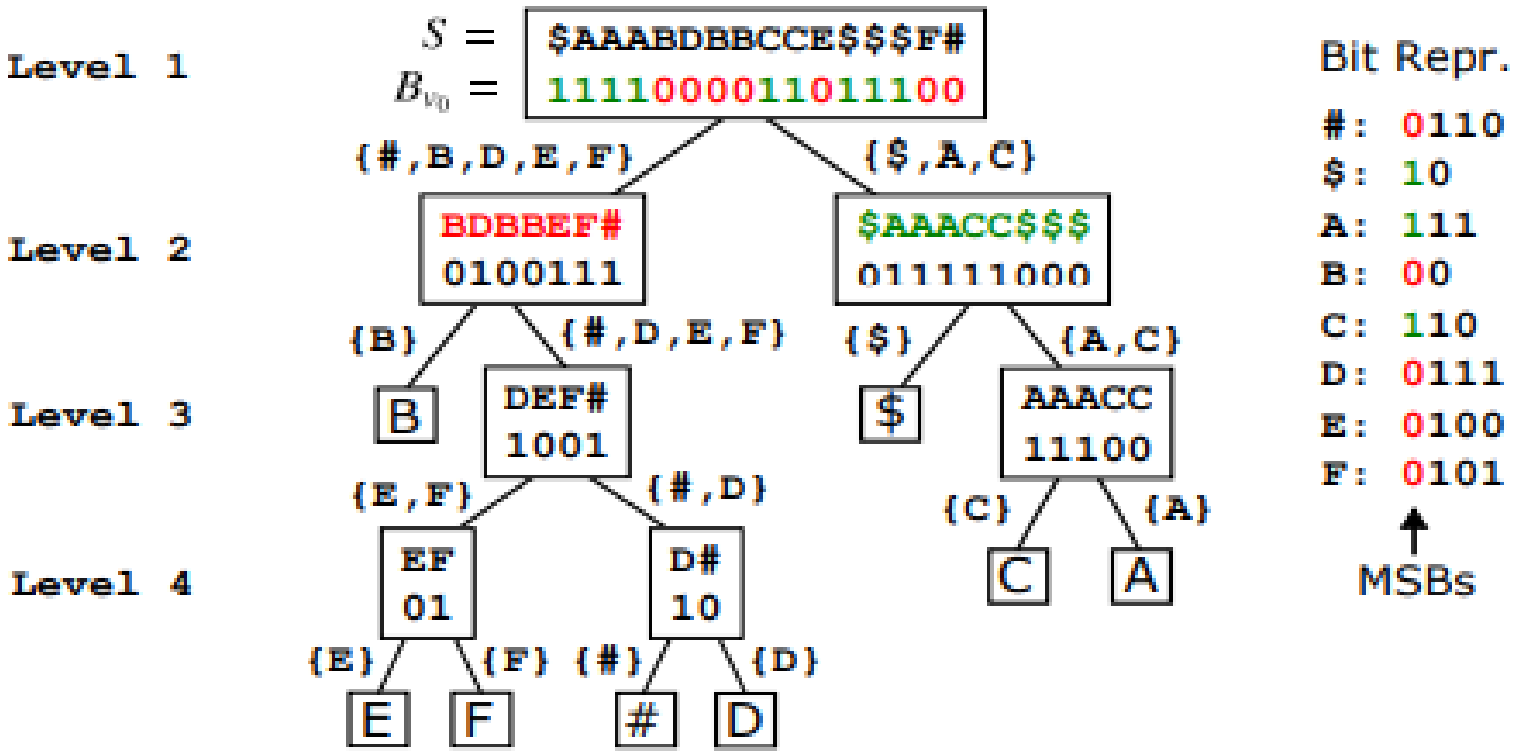
1 $w \leftarrow P[m-1]$; $sp \leftarrow C[w]$; $ep \leftarrow C[w+1]$
2 **for** $i \leftarrow 2$ *to* m **do**
3 $w \leftarrow P[m-i]$
4 $sp \leftarrow C[w] + rank_w(T_{bwt}, sp)$
5 $ep \leftarrow C[w] + rank_w(T_{bwt}, ep)$
6 **if** $sp \geq ep$ **then return** `NotFound`
7 **return** $[sp, ep)$

$$sp = C[\mathsf{B}] + 1 = 9$$

$$ep = C[\mathsf{B}] + 3 = 11$$

```
 9: BA$CB$DA$#FEBA$C      C
10: BA$CBA$CB$DA$#FE      E
```

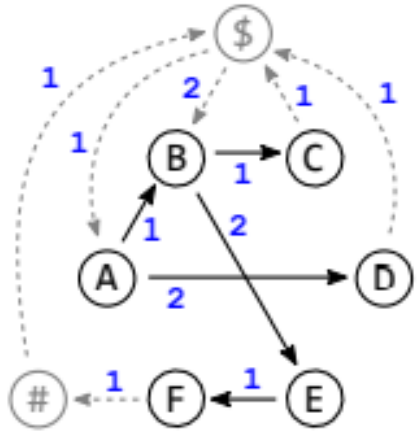
1.Huffman tree
2.Huffman-shaped wavelet tree(HWT)



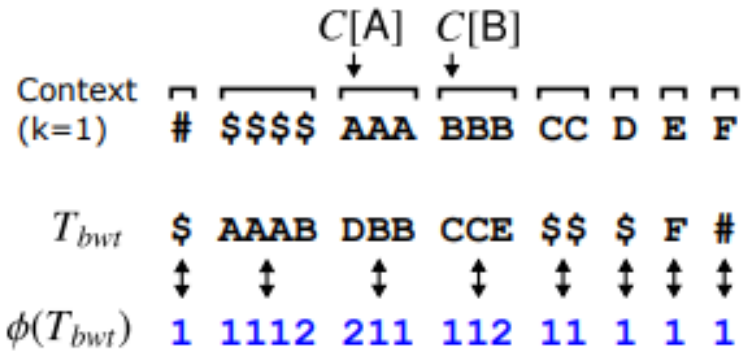
$$rank_{\mathsf{B}}(T_{bwt}, 5) = 1$$

Rank0(B, Tbwt, 5) = 1
Rank0(B, Tbwt, 1) = 1

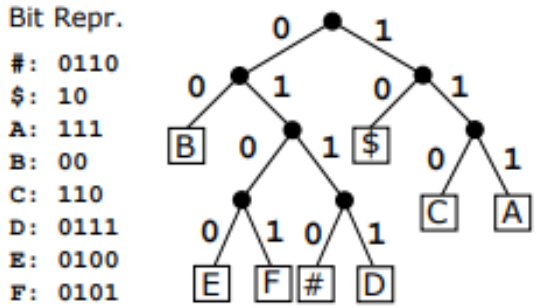
Relative movement labeling (RML)



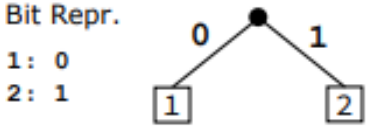
(a) ET-graph and RML



(b) Labeling T_{bwt} with RML



(a) T_{bwt}



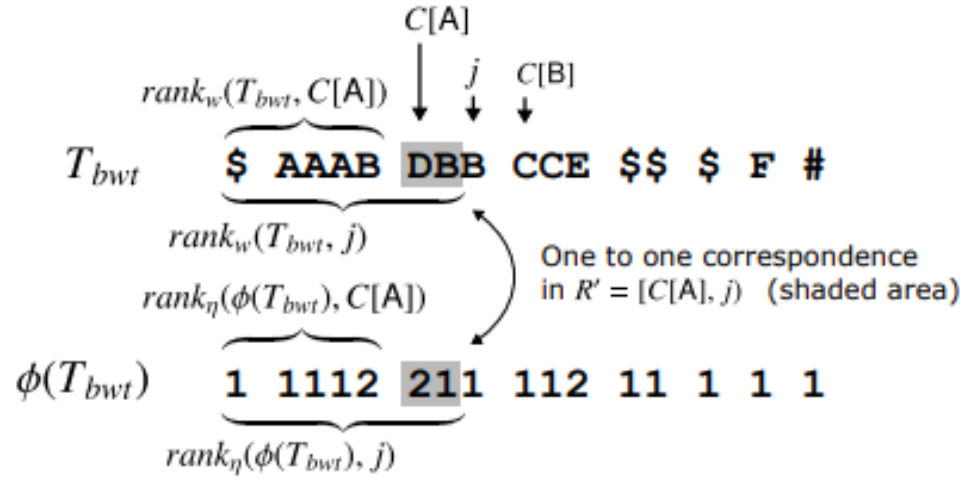
(b) $\phi(T_{bwt})$: CiNCT

Algorithm 2: Emulating $rank_w(T_{bwt}, j)$ by using only $\phi(T_{bwt})$ (*PseudoRank*$(\phi(T_{bwt}), j, w, w', Z_{w'w})$)

Input: Labeled BWT string of length n: $\phi(T_{bwt})$,
Location of rank j, Correction term $Z_{w'w}$,
Target symbol w, Previous symbol w'

Output: The value of $rank_w(T_{bwt}, j)$

1 **if** $w \in N_{out}(w')$ *and* $C[w'] \leq j \leq C[w'+1]$ **then**
2 $\eta \leftarrow \phi(w|w')$ // RML
3 **return** $rank_\eta(\phi(T_{bwt}), j) - Z_{w'w}$
4 **return** NotFound



$$rank_w(T_{bwt}, j) - rank_w(T_{bwt}, C[w']) = rank_\eta(\phi(T_{bwt}), j) - rank_\eta(\phi(T_{bwt}), C[w']). \quad (5)$$

$$Z_{w'w} := rank_\eta(\phi(T_{bwt}), C[w']) - rank_w(T_{bwt}, C[w']). \quad (7)$$

$$rank_w(T_{bwt}, j) = rank_\eta(\phi(T_{bwt}), j) - Z_{w'w}, \quad (6)$$

Algorithm 3: Finding the suffix range $[sp, ep)$ for a given query P of length m based on $\phi(T_{bwt})$ (*LabeledSearchFM*)

Input: Labeled BWT string of length n: $\phi(T_{bwt})$,
Query of length m: $P[0, m)$,
Correction terms: $\{Z_{w'w}\}$

Output: Range of T_{bwt} that matches to P

1 $w \leftarrow P[m-1]$; $sp \leftarrow C[w]$; $ep \leftarrow C[w+1]$
2 **for** $i \leftarrow 2$ to m **do**
3 $w' \leftarrow w$ `// Save the previous symbol`
4 $w \leftarrow P[m-i]$
5 **if** $w \notin N_{out}(w')$ **then**
6 **return** `NotFound`
7 $sp \leftarrow C[w] + \mathit{PseudoRank}(\phi(T_{bwt}), sp, w, w', Z_{w'w})$
8 $ep \leftarrow C[w] + \mathit{PseudoRank}(\phi(T_{bwt}), ep, w, w', Z_{w'w})$
9 **if** $sp \geq ep$ **then**
10 **return** `NotFound`
11 **return** $[sp, ep)$

$$T = \underbrace{\text{FEBA}}_{T_1^{\text{rev}}}\$\underbrace{\text{CBA}}_{T_2^{\text{rev}}}\$\underbrace{\text{CB}}_{T_3^{\text{rev}}}\$\underbrace{\text{DA}}_{T_4^{\text{rev}}}\$\#. \quad (1)$$

Algorithm 4: Extracting a sub-path $T[i-l, i)$ for given $j = ISA[i]$ and $l > 0$ (*extract*)

Input: Labeled BWT: $\phi(T_{bwt})$, Position on T_{bwt}: j, Extraction length: l, Correction terms: $\{Z_{w'w}\}$
Output: A substring $S := T[i-l, i)$

$w' \leftarrow BinarySearch(j, \{C[w']\})$ // T[i]
for $k \leftarrow 1$ to l **do**
  $\eta \leftarrow \phi(T_{bwt})[j]$; $w \leftarrow decode(\eta|w')$; $S[l-k] \leftarrow w$
  $j \leftarrow C[w] + PseudoRank(\phi(T_{bwt}), j, w, w', Z_{w'w})$
  $w' \leftarrow w$ // Save previous symbol
return S