

Implementing Rank Data Structure

Daanish Mahajan

Bitvector rank operation

- **Input:** A bitvector is an array $B[1 \dots n]$ of boolean values, i.e., $B[i] \in \{0,1\}$
- **Objective:** Design a space-efficient data structure to support queries of the following type
 - $rank_1(B)[i] = \left| \{i' \mid 1 \leq i' \leq i, B[i'] = 1\} \right|$
 - $rank_0(B)[i] = \left| \{i' \mid 1 \leq i' \leq i, B[i'] = 0\} \right|$
 - NOTE: $rank_0(B)$ is directly known from $rank_1(B)$
- An important building block of space-efficient data structures for general data as well as genomic data indexing

B
0
1
0
1
1
0
1
1
0

Naive Algorithm

B

0	1	1	0	1	0	1	1	0	0	0	1	1	1	0	1	0	1	0	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$rank_1(B)[i]$

0	1	2	2	3	3	4	5	5	5	5	6	7	8	8	9	9	10	10	11	11	12	13	13
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Succinct Algorithm (4-Russian technique)

