LIST INTERSECTION: 1) Mutual partitioning 2) Doubling

S_1 = 1 2 5 7 10 16 20

S_2 = 1 8 10

1) MUTUAL PARTITION:

- pick middle element of the shortest list
- binary search of 8 in the other list: if don't match => NO ELEMENTS REMOVED
- partition S_1 & partition S_2 => RECURSE

S_1 = 1 2 5 7 ↑ 10 16 20

S_2 = 1 [8] 10

$S_1 \Rightarrow S_{11}$ = 1 2 5 7 ; S_{12} = 10 16 20

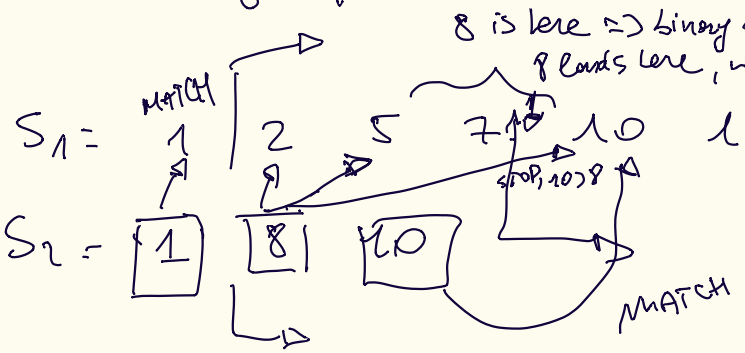
$S_2 \Rightarrow S_{21}$ = [1] (MATCH) ; S_{22} = [10] (MATCH)

3 binary searches

- values returned: 1, 10 => $S_1 \cap S_2$ ✓

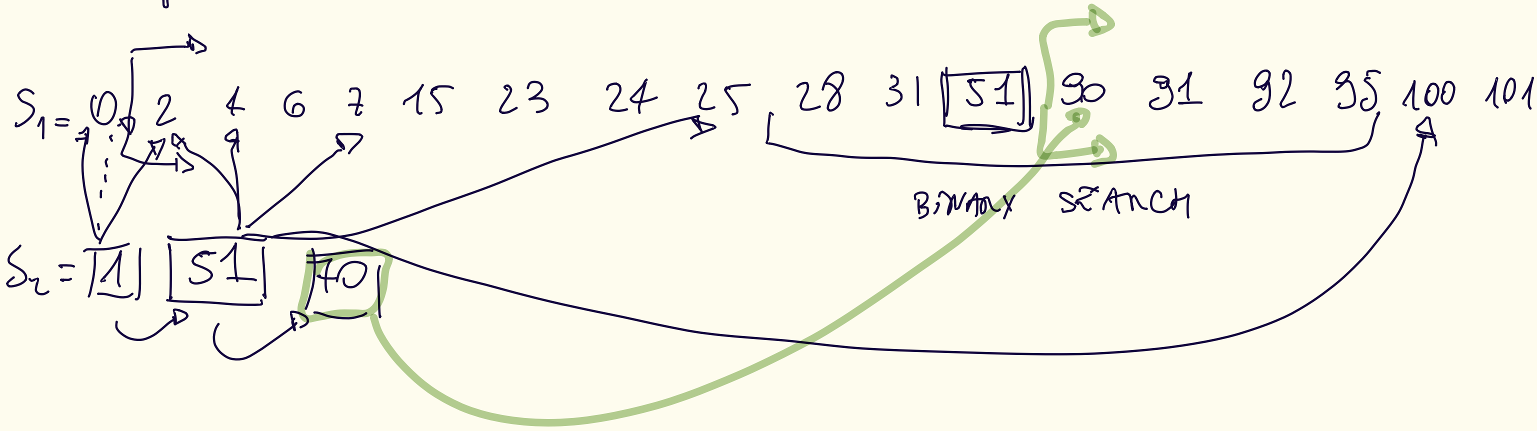
Doubling search

- scan the shorter list
- do the jump in the longer one (search the element in the other list comparing from the first element: $2^0, 2^1, 2^2, 2^3, 2^4$) when match => discard the position matched



=> RETURNED: 1, 10

Doubling search



MUTUAL PARTITION

S_1 = 1 2 5 7 10 16 20

S_2 = 1 8 10

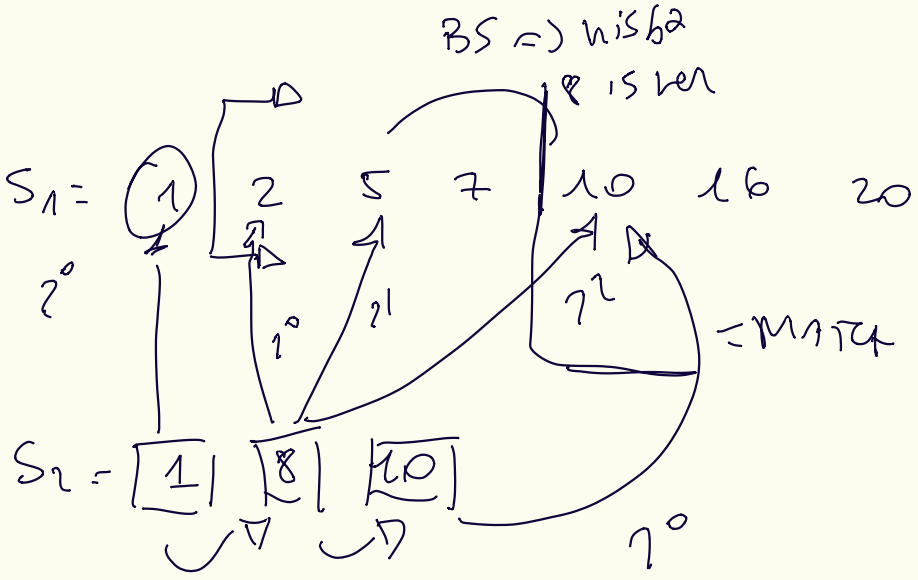
1 2 5 7 10 16 20

1 => smaller => 1 matched

10 => smaller, 10 matched

$S_1 \cap S_2 = \{1, 10\}$

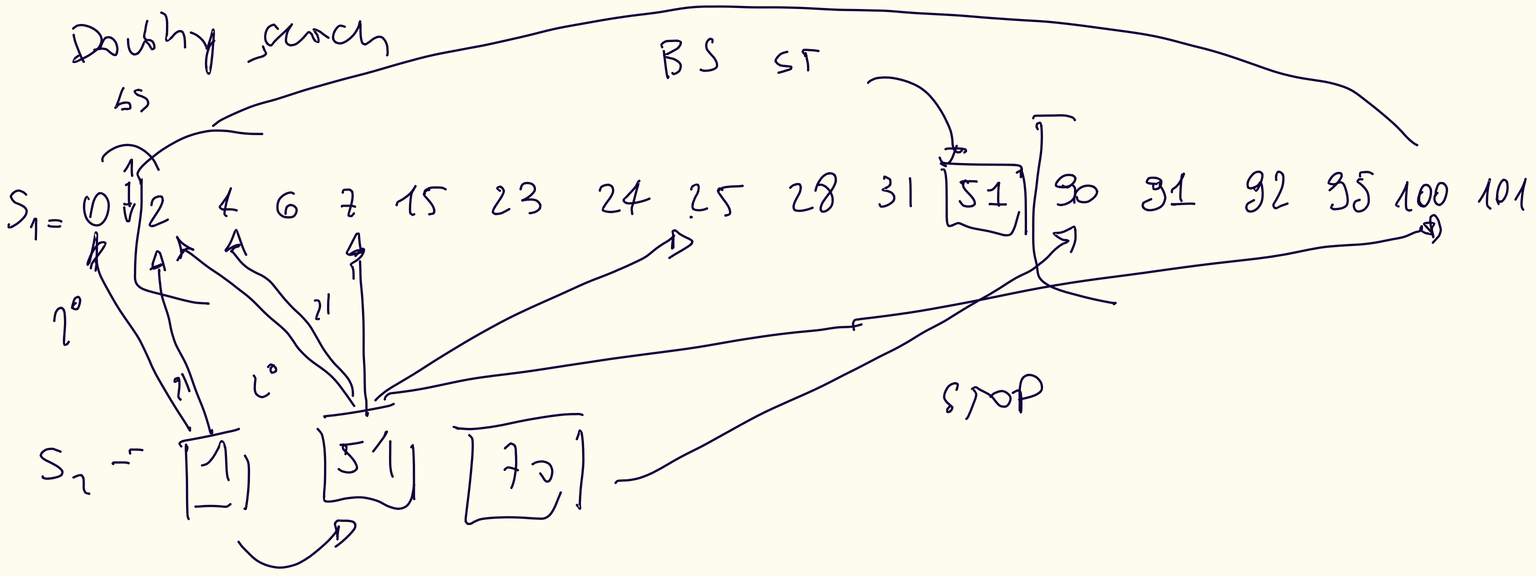
Doubly search



RETURN =

1, 10

Doubling search



Return: 51