

Maximize Number of 1's

Brute Force Time - $O(N^2)$, Space - $O(N)$

eg. 1 0 0 1 1 0 1 0 1 1 1

① Create a count func to identify max. no. of contiguous array.

② Iterate 2 loops (nested) such that convert $arr[j] = 1$ (those which are zero) & push into queue to revert back the operation.

Do this for $i = 0 \rightarrow n$ & $j = i \rightarrow n$

Optimized

use sliding window technique.

initialize $i = -1$ & $j = 0$ for loop with $j = 0$ to n .

iterate ~~until~~ until count of 0 $> m$.

1 0 0 1 1 0 1 0 1 1 1

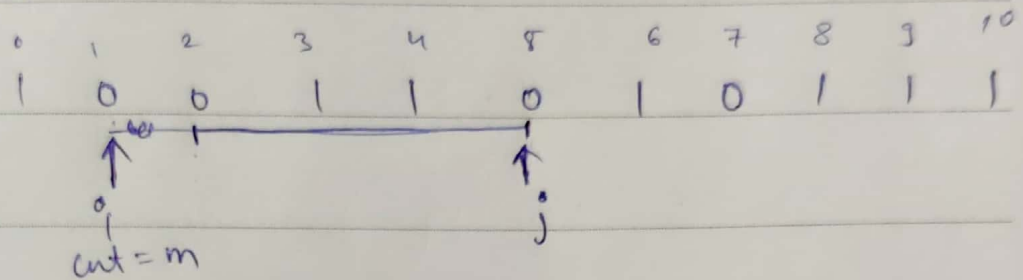
$i = -1 \uparrow$

$\uparrow j$

②

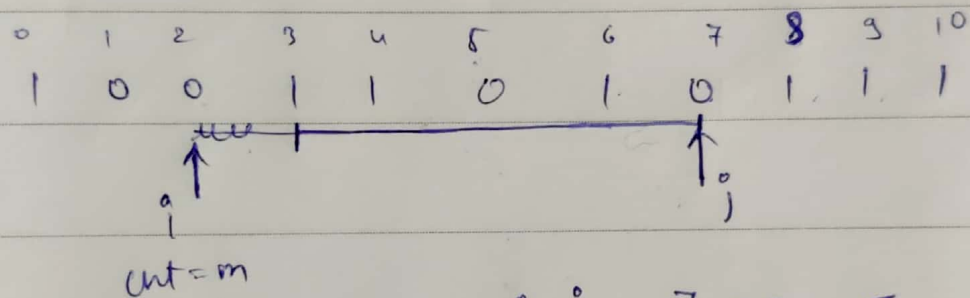
Now, our count has become $> m$

So, forward i until my cnt again become $= m$.



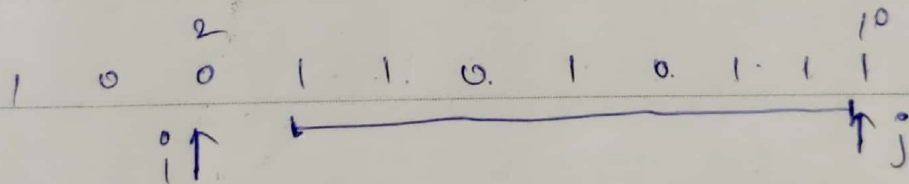
$$ans = j - i = 5 - 1 = 4$$

③



$$ans = j - i = 7 - 2 = 5$$

③



$$ans = 10 - 2 = 8$$