

Cpts 350

HW-4

Abhilash Ambati

11602294

1. a.) the constrain on the pointers are onepass, In-place and liner-time.

onepass should traverse each element ~~that~~ once. In-place should not allocate any extra memory to complete the task. and liner-time should have $O(n)$ time complexity.

b.) for the algorithm to compile with these constraints we have to make sure that ~~the~~ ~~each~~ ~~element~~ the pointers pass each element only once and ~~the~~ each pointer moves in a different direction.

to accomplish this we are naming one pointer as left pointer and the other one as right pointer. the left pointer will be on the first index and the right pointer will be on the last index.

Now we are going to make each ~~pointer~~ pointer move towards each other until they ~~or~~ reach ~~to~~ halfway.

~~on~~ In this process if the left pointer is purple and right is brown they are going to be swapped.

1 = purple
2 = brown.

c.) Part_1(A[], int n)

{
 int L = 0, R = n-1;

 while (L < R)

 {

 if (A[L] == 1) {

 L++;

 }

 if (A[R] == 2) {

 R--;

 }

 if (A[L] == 2 && A[R] == 1) {

 swap(L, R);

 }

 }

}

return A;

}

2.)

part-2 (A[], int n)

{

int S = 0, m = 0, E = n - 1;

while (mid <= end)

{

if (A[m] == brown) {

S++;

}

m++;

} ~~m++;~~

else if (A[m] == black) {

swap(m, E);

}

else {

m++;

}