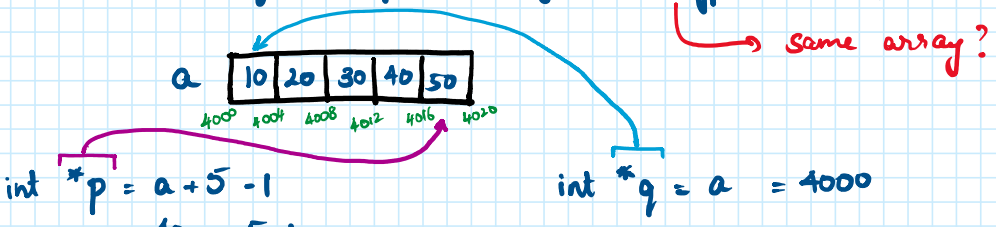
2. Pointer Arithmetic, Binary Search, Bubble Sort

05 March 2024 08:51

Pointer Arithmetic

- Addition of integer to pointer → possible
- Subtraction of integer from pointer → possible
- Addition of two pointers → NOT possible
- Subtraction of two pointers of same type → possible
  - → same array?

a+1 | a-1
vs
a++ | ++a a--



int *p = a+5-1
= 4000+5-1
= 4020-1
= 4016

int *q = a = 4000

p-q = (a+4)-a
= 4 → difference in addresses; returns ans as an integer

*p++ , *(p++), (*p)++ } Put these in a loop and see what happens

NOTE:

a → base address
a+1 → base address +1
a[0] → same as base address
&a+1 → considers &a as entire array

count of odd no. and even no. in array

Binary Search

Values should be sorted for this to work.

given some value "key" to search for, return its index

low = 0, high = 7 [n-1]

mid = (0+7)/2 = 3

Check: key == a[mid]

false → key < a[mid]

true → return mid

false → Check upper half of array → Change "low" value to lower bound of upper half and repeat the same process [high = mid-1]

true → Check lower half of array → Change "high" value to upper bound of lower half and repeat the same process [low = mid+1]

```
while (low <= high)
{
    mid = (low + high)/2;
    if(key == a[mid])
        return mid;
    else if (key < a[mid])
        high = mid-1;
    else
        low = mid+1;
}
return -1
```

BUBBLE SORT

Pair by pair swapping based on some given condition

Example: Sorting given array into ascending order

a[0]	30	20	20	20	20	20	20	20	10	10	10
a[1]	20	30	30	30	30	30	10	10	20	20	20
a[2]	50	50	50	10	10	10	30	30	30	30	30
a[3]	10	10	10	50	40	40	40	40	40	40	40
a[4]	40	40	40	40	50	50	50	50	50	50	50