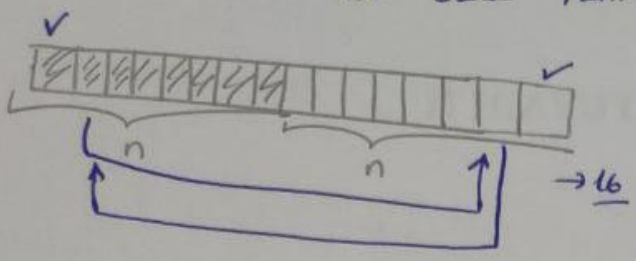
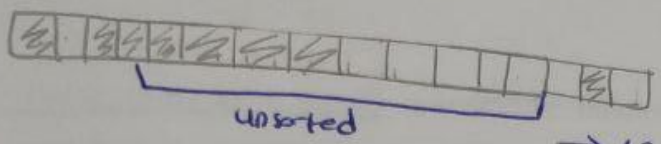
CSE 321 Homework 3

Q1)

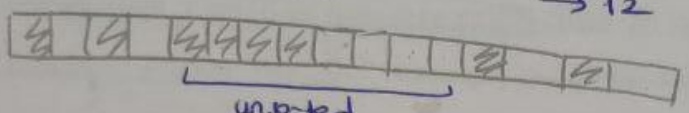


our first move is changing 2nd box and (2n-1)nd

→ 16



→ 12



→ 8

decrease by two

16 − 12 − 8 → reducing by $\frac{n}{2} = \frac{8}{2} = 4$

→ if n is odd → 17 − 13 − 9 ($\frac{n-1}{2}$)

When we apply this on recurrence our function will be

$m(n) = m(n-2) + 1$ for $n > 2$, $m(2) = 1$

$m(1) = 0$

We must make movements at least $\frac{n}{2}$ → best case $O(\frac{n}{2}) \Rightarrow O(n)$

→ worst case

a2) Finding fake coin

Suppose we divide the coins into 3 port. (at least two port contain some number of coins)
Now we can eliminate equal weighted ports. And remain 1/3 of coins.

```
If n=1
    return fake coin
else
    divide 3 ports (A,B,C) the coins.
    weigh A, B
    if A and B equal
        recursive with C
    else
        recursive with lighter of A or B
```

our size reducing with size * 1/3

this means $O(n) = \log_3 n$

Best case ⇒ $n = 1$ constant

$O(n) = 1$

worst case $O(n) = \log_3 n$

↳ but it changes our factor.
if we choose 2 as factor
our worst case $O(n) = \log_2 n$

a3) Quick Sort Vs Insertion Sort

quick sort → for 5 elements ⇒ 3
insertion sort → for 5 elements ⇒ 7

arr [12, 11, 13, 5, 6]

quick faster than iserton sort

recursive with C

else

recursive with lighter of A or B

Best case ⇒ $\frac{n=1}{2}$ con

$O(n) = 1$

worst case $O(n) = \log_3 n$

↳ but it changes our

if we choose 2 as fa

our worst case $O(n) =$

(q 3) Quick Sort Vs Insertion Sort

quick sort → for 5 elements ⇒ 3, 7 — arr [12, 11, 13, 5, 6] — quick faster than insertion sort

insertion sort → for 5 elements ⇒ 7, 7, 7

Quick → for 7 elements → 5, 7

insertion → for 7 elements → 11, 7 — arr [10, 7, 8, 9, 1, 5, 20] — quick faster than insertion sort

Insertion sort average case analysis

The worst case will occur if array sorted by decreasing order. To insert the last element we need at most (n-1) comparisons, insert to second last element (n-2) ---- so on.

therefore: $2 \times (1 + 2 + \cdots n-2 + n-1)$

To calculate recurrence

$$\sum_{q=1}^{p} q = \frac{p(p+1)}{2} \Rightarrow \frac{2(n-1)(n-1+1)}{2} \Rightarrow \frac{O(n^2)}{5}$$

quicksort

Best case recurrence is ⇒ $T(n) = 2T\left(\frac{n-1}{2}\right) + O(n)$ → with master theorem ⇒ $T(n) = O(n \log n)$

worst case ⇒ $T(n) = T(n-1) + O(n)$ → with master → $T(n) = O(n^2)$

Average case analysis ⇒ $T(n) = O(n \log n)$

we must make mo

Q4) Finding median

Steps ⟹ select a pivot we divide array 2 part. (greater than pivot and less than pivot)
and we are looking for the Kth smallest we continue the partition when K is null.

Average complexity ⇒ $n + 1/2\,n + 1/4\,n - \cdots < 2n \sim n = O(n)$

Worst case ⇒ if pivot element will be either the largest or the smallest element in the array, occur the worst case.
So we can divide the array just one array with size $(n-1)$
so recurrence equation is $T(n) = T(n-1) + n$

$\Rightarrow O(n^2)$

Q5) Exhaustive search

First we must find all subsets in our arr. Then we must calculate the given formula
That algorithm is subexponential so our time complexity is $O(2^n)$ in worst case.