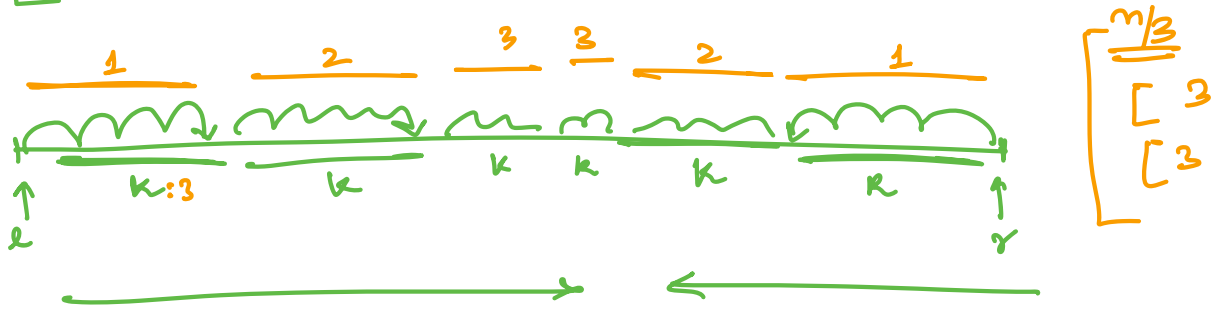
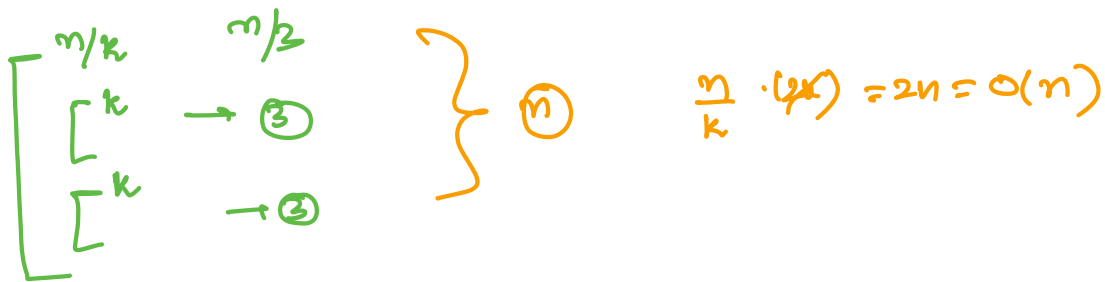


Time Complexity of Quick Sort

Partitioning time + 2 Rec call



Worst Case:

Pivot Element: Smallest / Largest



$$T(n) = n + T(n-1) + T(1)$$

$$T(n) = T(n-1) + \underline{T(1)} + n$$

$$T(n) = T(\cancel{n-1}) + n + 1$$

$$T(\cancel{n-1}) = T(\cancel{n-2}) + (n-1) + 1$$

$$T(\cancel{n-2}) = T(\cancel{n-3}) + (n-2) + 1$$

$$T(\cancel{1}) = 1$$


$$\begin{aligned}
 T(n) &= \underbrace{n + (n-1) + (n-2) + \dots + 0 + n}_{\text{}} \\
 &= n \left(\frac{n+1}{2} \right) + n \\
 &= \frac{n^2}{2} + \frac{n}{2} + n \\
 &= O(n^2)
 \end{aligned}$$

Best Case: Pivot Element: Median

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + n \quad \rightarrow \text{Same as merge sort}$$

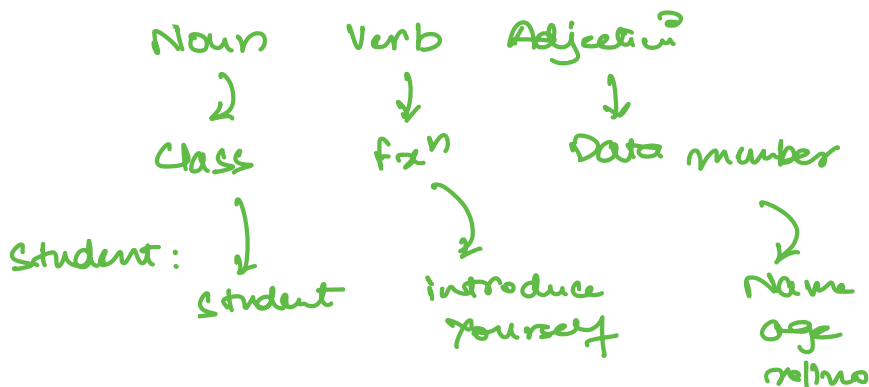
$$T(n) = O(n \log n)$$

Data Structures

Array:  $O(1)$ time

Stack ??

- C language - Procedural (fxⁿ)
- C++ language - object oriented

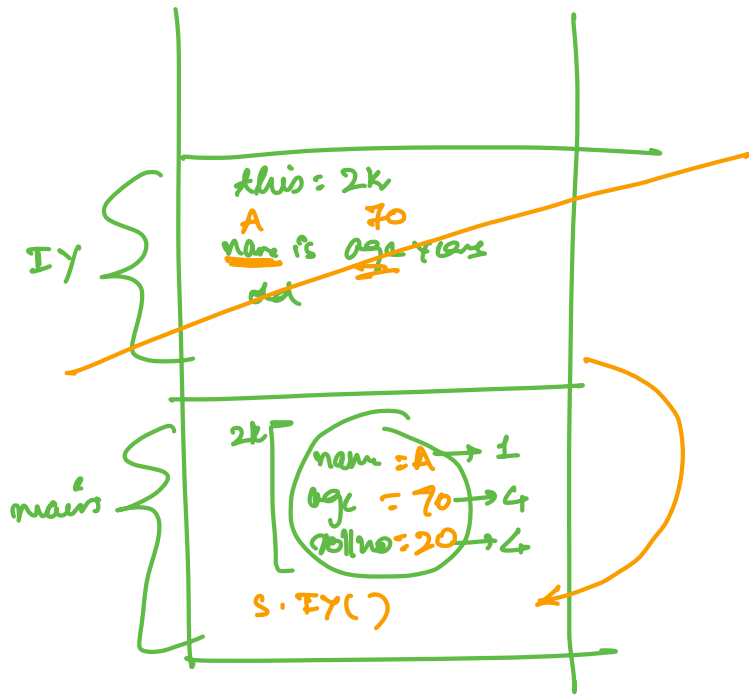


```

class Student
{
    char name;
    int age;
    int rollno;
}

```

Template
no memory is allocated



```

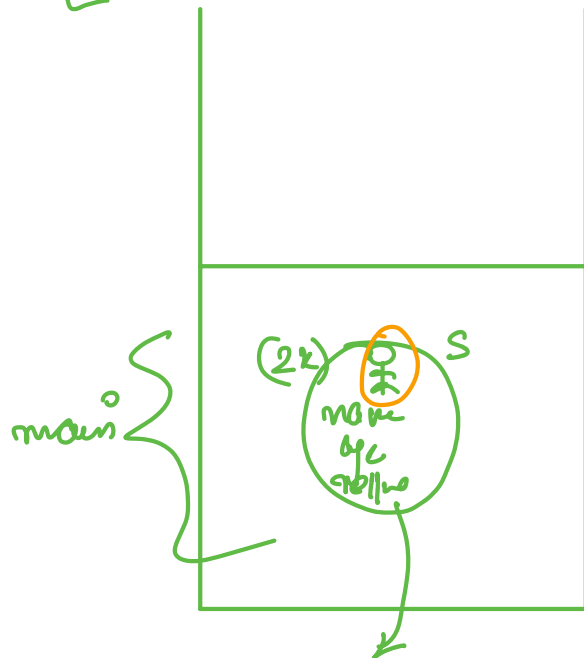
main
[ Student s ;

```

```

vector<int> v;

```



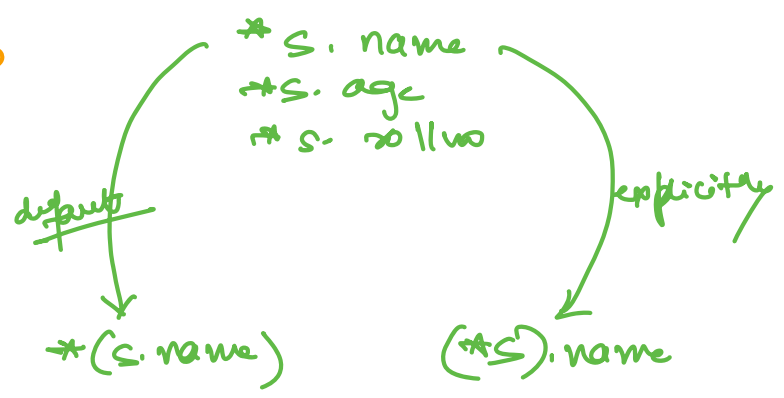
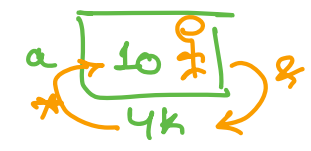
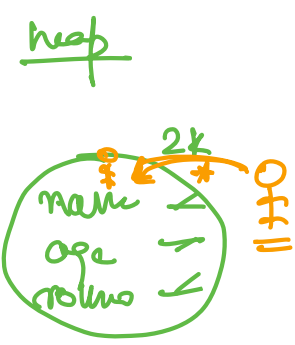
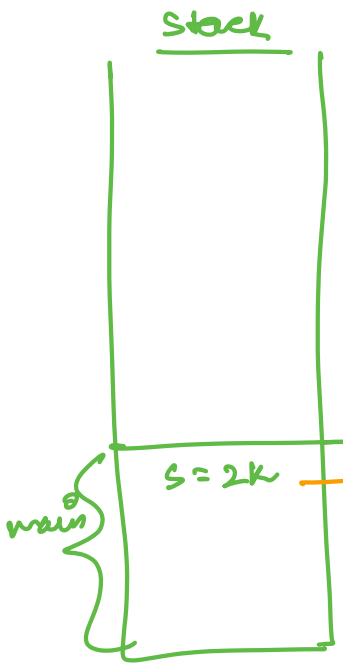
① memory stack

② inside memory location

• access data members

```
main
Student *s = new Student()
```

```
vector<int> v = new vector();
```



s = address store

(*s).name
s -> name

dot

