# Assignment - 1

① What is the time complexity of the following code?

```
int a=0;
for (i=0; i<N; i++)
{
    for (j=N; j>i; j--)
    {
        a = a+i+j;
    }
}
```

Sol:

```
int a=0;                          → 1
for (i=0; i<N; i++)               → n
{
    for (j=N; j>i; j--)           → n(n-i)
    {
        a = a+i+j;                → n(n-i)
    }
}
```

$T_n = 1 + n + n^2 - nk_1 + n^2 - nk_1$

$T_n = 2n^2 + (1-2k_1)n + 1$

Drop the non-dominant terms.

$T_n = 2n^2$

Drop the constant coefficients.

$T_n = n^2 = O(n^2)$

② Let us say you need to double an array size whenever the array is full. What is the amortized cost of operations involved?

Sol:- When array is not full, adding an element take

$O(1)$

When array is is full, doubling its size involves

i) Allocating a new array of double the size.

ii) Copying all existing elements from the old array to the new array. This copying operation take $O(n)$, where $n$ is the size of the array at the time of resizing.

Amortized Cost:-

Cost of each resizing operation is proportional to the size of the array being copied.

- Consider $n$ operations on the array.

Resizing occurs at array sizes

$1, 2, 4, 8, \ldots, 2^k$ where $2^k \leq n$

Total cost $(T) = 1 + 2 + 4 + 8 + \ldots + n$

$T = 2n - 1$

$T \approx 2n$

Amortized cost $= \frac{T}{n}$

$= \frac{2n}{n}$

$= 2$

∴ Amortized cost of appending an element to an array that

doubles in size when full is $O(1)$.

③ Show that if fewer than $\lceil n/2 \rceil$ unions are performed, then atleast one set with a single element in it remains.

Sol: Consider $n$ sets with each containing a distinct element.

Each union operations reduces the total number of sets by 1. After $k$ union operations, the no. of sets remaining will be $n-k$.

If fewer than $\lceil n/2 \rceil$ union operations are performed then

$$k < \lceil n/2 \rceil.$$

$\therefore$ no. of sets remaining.

$$n-k > n - \lceil n/2 \rceil$$

$n - \lceil n/2 \rceil$

If $n$ is even;

$$\lceil n/2 \rceil = n/2 \text{ so } n - n/2 = n/2$$

If $n$ is odd,

$$\lceil n/2 \rceil = (n+1)/2$$

$$n - \lceil n/2 \rceil = n - (n+1)/2$$

$$= (n-1)/2$$

In either case, $n-k > n - \lceil n/2 \rceil$ means more than one set remains. Since there are still multiple sets, at least one of these sets must still contain a single element.

$\therefore$ If fewer than $\lceil n/2 \rceil$ union operations are performed, at least one set with a single element remains.

(4) Show the resulting sequence of operations using Union by Rank. Union(1,3), Union(2,4), Union(3,5), Union(1,7), Union(3,7), Union(6,9), Union(8,10) with link and rank arrays.

Sol: link[]: Points to the parent of each element in the disjoint set forest

rank[]: Tracks the rank of the tree rooted at each element.

Initially, each element is its own parent and has a rank of 0.

link: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

rank: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Operations:

1. Union(1,3):

→ Both are its own roots

→ Both ranks are 0.

→ Arbitarily make 1 as root and link 3 to 1 and increase the rank of 1 by 1.

link: [1, 2, 1, 4, 5, 6, 7, 8, 9, 10]

rank: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

2. Union(2,4):

→ Both 2 and 4 are their own roots.

→ Both ranks are 0.

→ Arbitarily make 2 as root, link 4 to 2 and increase the rank of 2 by 1.

link: [1, 2, 1, 2, 5, 6, 7, 8, 9, 10]

rank: [1, 1, 0, 0, 0, 0, 0, 0, 0, 0]

3. Union (3,5):

→ Root of 3 is 1 and 5 is its own root.

→ Rank of 1 is higher than rank of 5

→ link 5 to 1.

link: [1, 2, 1, 2, 1, 6, 7, 8, 9, 10]

rank: [1, 1, 0, 0, 0, 0, 0, 0, 0, 0]

4. Union (4,6):

→ Root of 4 is 2 and 6 is its own root

→ Rank of 2 is 1 which higher than rank of 6 is 0.

→ link 6 to 2.

link: [1, 2, 1, 2, 1, 2, 7, 8, 9, 10]

rank: [1, 1, 0, 0, 0, 0, 0, 0, 0, 0]

5. Union (7,8):

→ 7 and 8 are their own roots

→ Both ranks are 0.

→ Arbitrarily make 7 as root, link 8 to 7 and then increase the rank of 7 by 1

link: [1, 2, 1, 2, 1, 2, 7, 7, 9, 10]

rank: [1, 1, 0, 0, 0, 0, 1, 0, 0, 0]

6. Union (9,10):

→ 9 and 10 are their own roots

→ Both ranks are 0.

→ Arbitrarily make 9 as root, link 10 to 9 and the increase the rank of 9 by 1.

link: [1, 2, 1, 2, 1, 2, 7, 7, 9, 9]

rank: [1, 1, 0, 0, 0, 0, 1, 0, 1, 0]

7. Union (5,7):

→ Root of 5 is 1 and root of 7 is 7.

→ Both ranks of 1 and 7 are 1.

→ Arbitarily make 1 as root and link 7 to 1, then increase the rank of 1 by 1.

link : [1, 2, 1, 2, 1, 2, 1, 7, 9, 9]

rank : [2, 1, 0, 0, 0, 0, 1, 0, 1, 0]

8. Union (6,9):

→ Root of 6 is 2 and root of 9 is 9.

→ Both rank of 2 and 9 are 1.

→ Arbitarily make 2 as root, link 9 to 2 and increase the rank of 2 by 1.

link : [1, 2, 1, 2, 1, 2, 1, 7, 2, 9]

rank : [2, 2, 0, 0, 0, 0, 1, 0, 1, 0]

9. Union (8,10):

→ Root of 8 is 7 and Root of 10 is 9.

→ Both ranks of 7 and 9 are 1.

→ Arbitarily make 7 as root and link 9 to 7 and increase the rank of 7 by 1.

link : [1, 2, 1, 2, 1, 2, 1, 7, 7, 9]

rank : [2, 2, 0, 0, 0, 0, 2, 0, 1, 0]