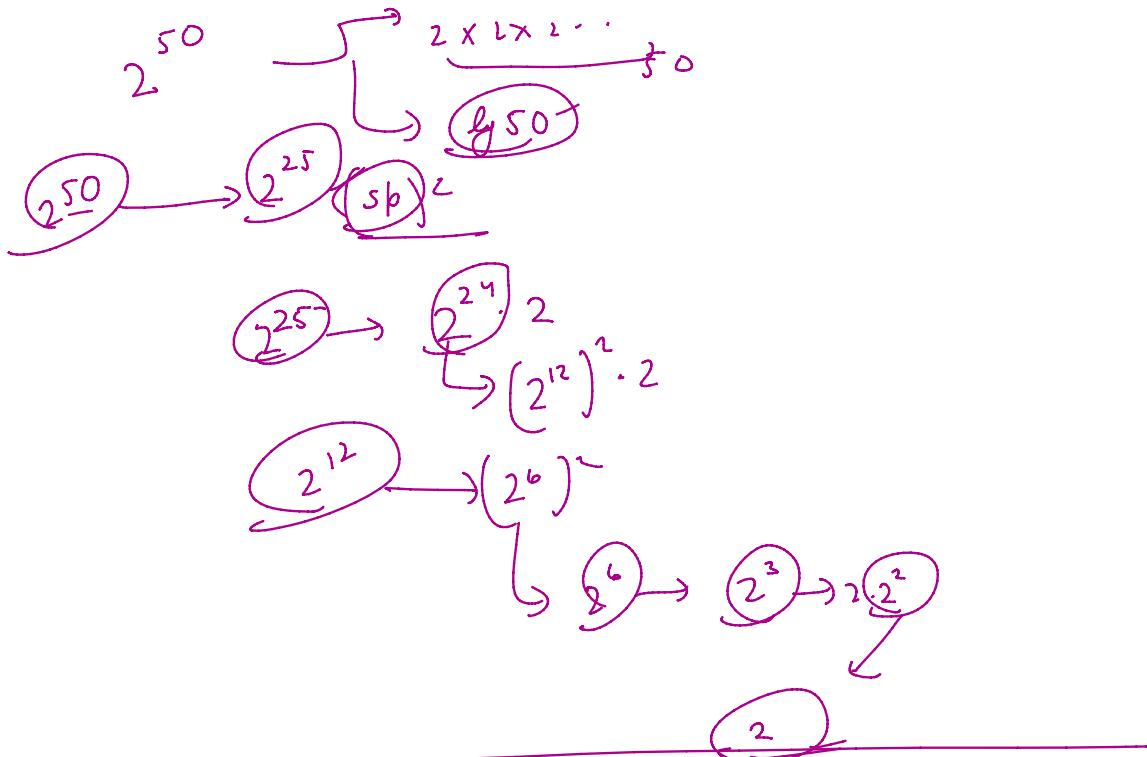


Recursion

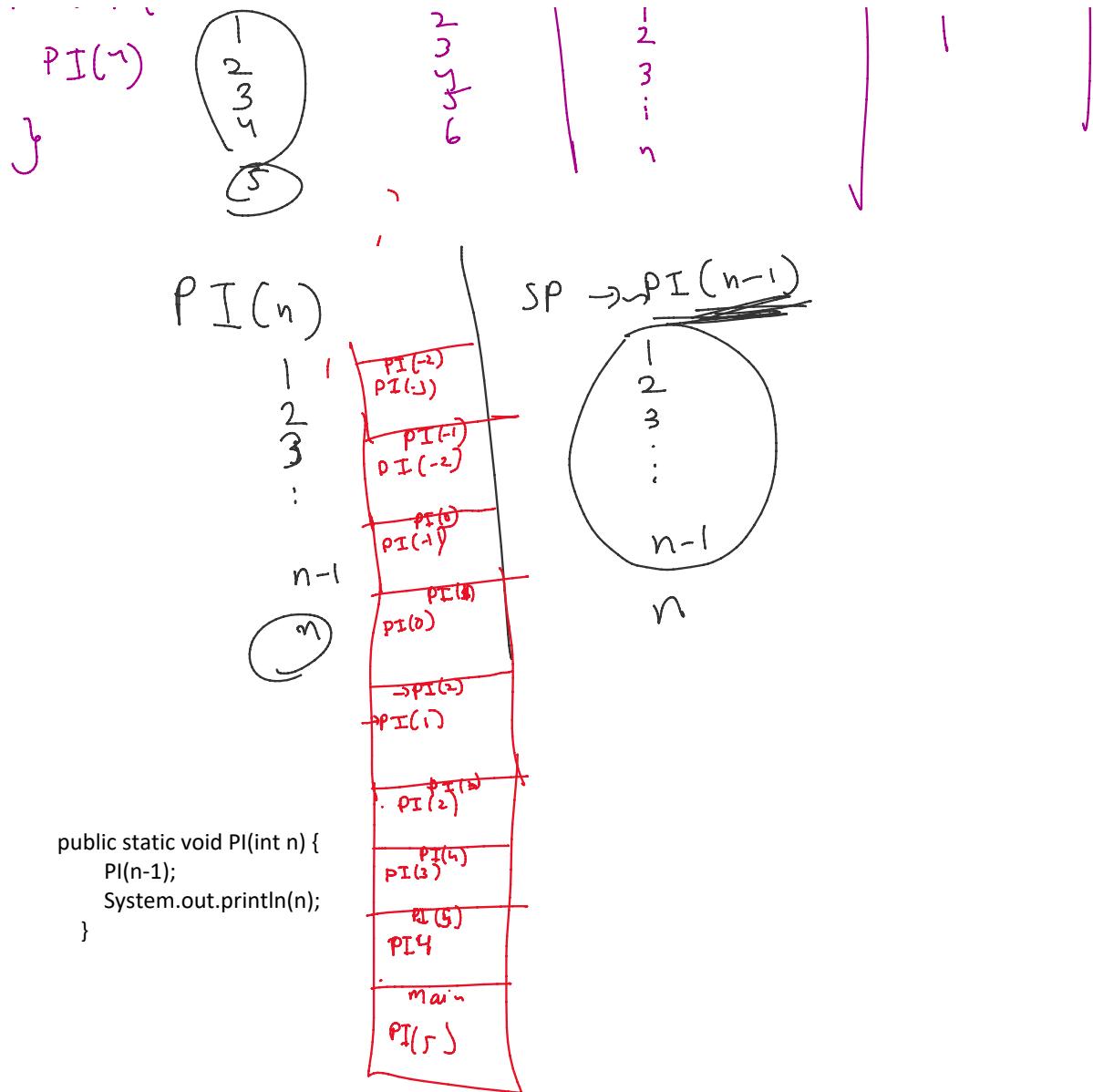
07 May 2023 10:19



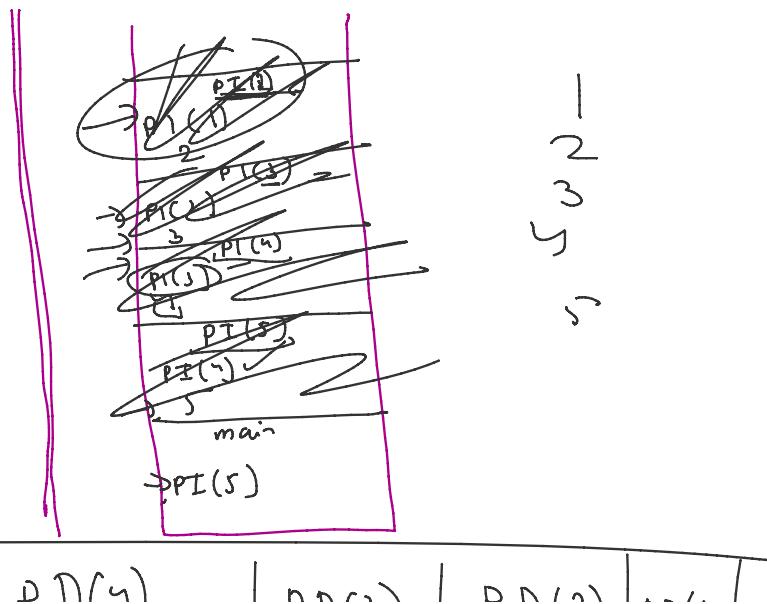
- 1) Identify Biggest/Bigger Problem
 - 2) Identify Smaller Problems
 - 3) Assume smaller works
 - 4) Create Bigger solⁿ wif smaller solⁿ
- 5) Smallest Problem identify

PI	1	PI(5)	PI(4)	PI(3)	PI(2)
	2		1 2 3 4 5	1 2 3	1 2
	3				
	4				
	5				

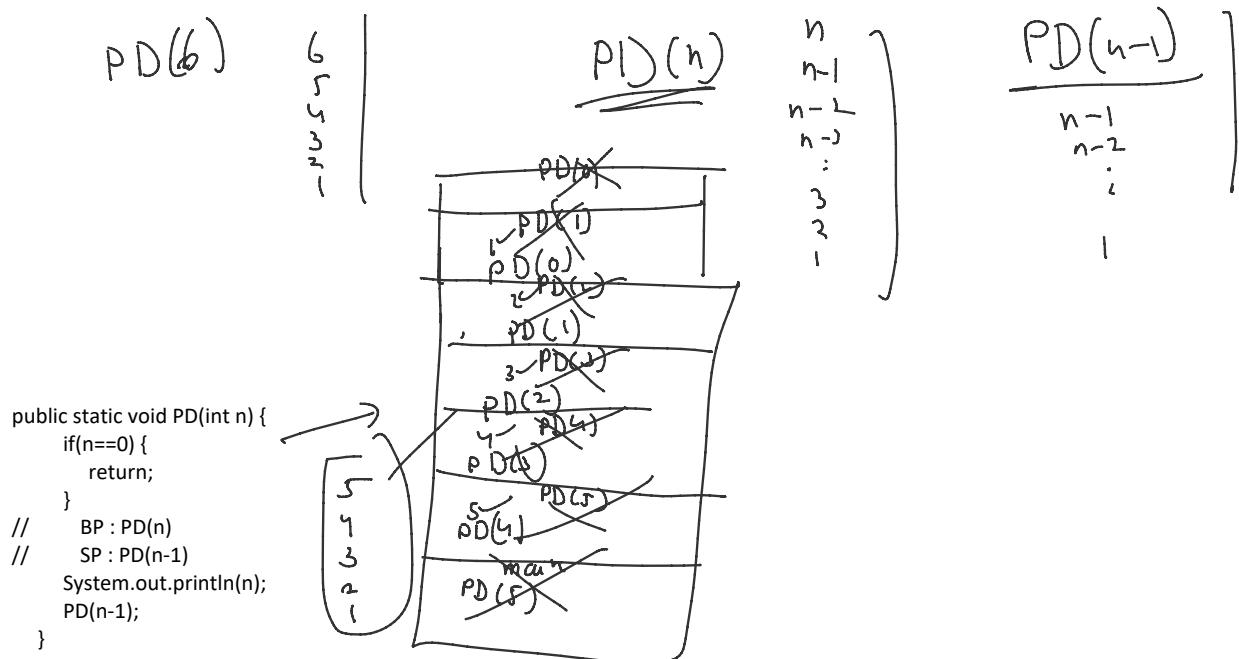
PI(5)	5	PI(6)	PI(n)	PI(1)	PI(0)
	1 2 3 4 5	1 2 3 4 5	1 2 3	1	
PI(1)	1 2 3 4 5				



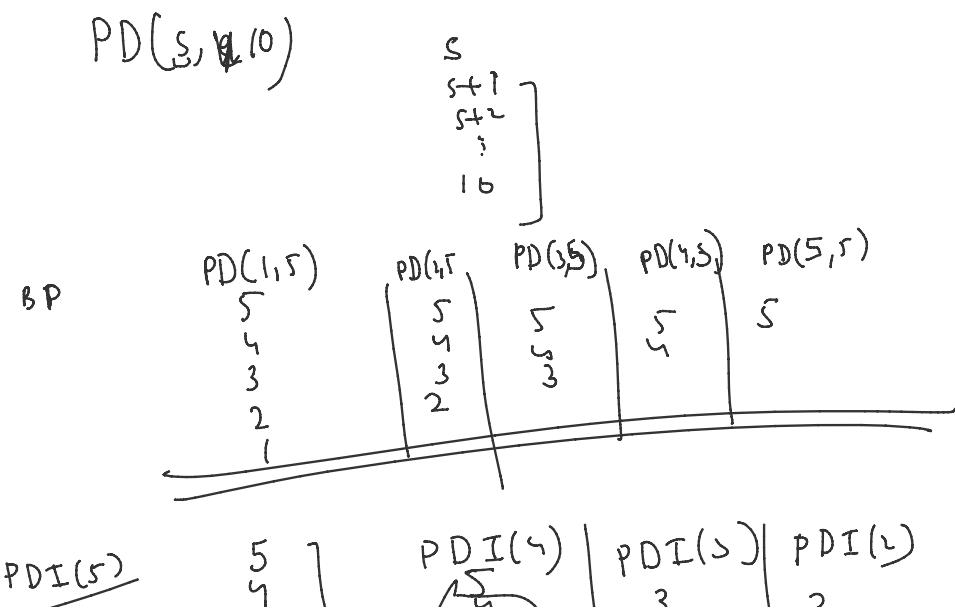
```
public static void main(String[] args) {  
    PI(5);  
}  
  
public static void PI(int n) {  
    if(n==1) {  
        → System.out.println(1);  
        return;  
    }  
    //    BP : PI(n)  
    //    SP : PI(n-1)  
    PI(n-1);  
    System.out.println(n);  
}
```

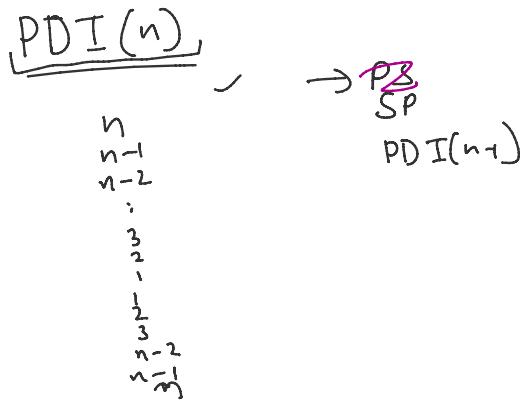
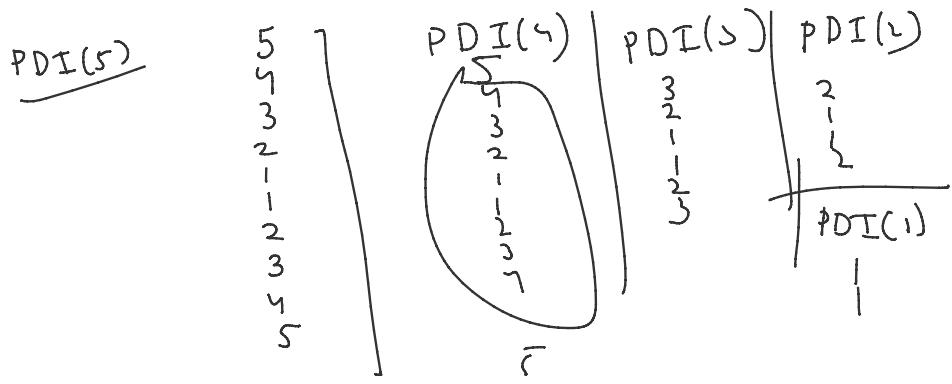


	PD(5)	PD(4)	PD(3)	PD(2)	PD(1)	X
	5 4 3 2 1	4 3 2 1	3 2 1	2 1	1	

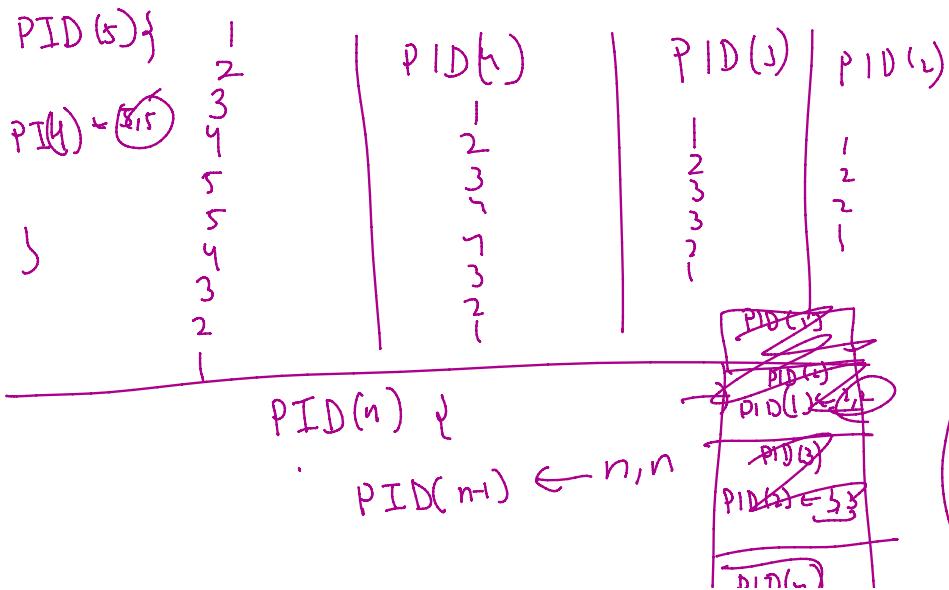
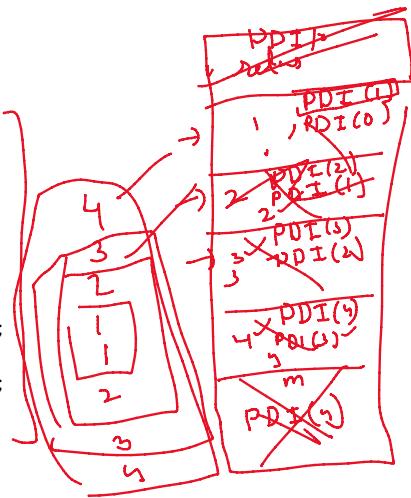


n = 5	BP(1, 5) (2, 5)	(3, 5)	(4, 5)	(5, 5)	(6, 5)
	2 3 1 5	2 3 1	3 1	4 1	5





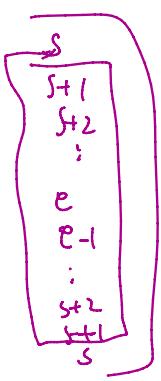
```
public static void PDI(int n) {
    if (n == 0) {
        return;
    }
    // BP : PDI(n)
    // SP : PDI(n-1)
    System.out.println(n);
    PDI(n - 1);
    System.out.println(n);
}
```



$BPC(1, 5)$

1
2
3
4
5
5
5
3
2
1

s, e



$SP \rightarrow (s+1, e)$

s

L, S, E

2
3
4
5
5
3
2

3, S, E

3
4
5
5
5

4, S, E

4
5
5
5

5, S, E

5
5
5

push $SP \leftarrow s$

