

1 More, MORE, MOREEEE (Spring 2016, MT2)

For all the methods below, give the runtime in $\Theta(\cdot)$ notation as a function of N . Your answer should be simple, with no unnecessary leading constants or summations.

```

1 public static void p1(int N) {
2     for (int i = 0; i < N; i += 1) {
3         for (int j = 1; j < N; j = j + 2) {
4             System.out.println("hi !");
5         }
6     }
7 }

```

N
 $N / 2$
 $N (N / 2) = N^2 / 2 \sim N^2$
 (N^2)

P1 answer: $\Theta(\quad)$

```

1 public static void p2(int N) {
2     for (int i = 0; i < N; i += 1) {
3         for (int j = 1; j < N; j = j * 2) {
4             System.out.println("hi !");
5         }
6     }
7 }

```

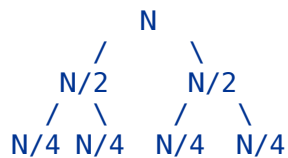
N
 $\log N$
 $(N \log N)$

P2 answer: $\Theta(\quad)$

```

1 public static void p3(int N) {
2     if (N <= 1) return;
3     p3(N / 2);
4     p3(N / 2);
5 }

```



P3 answer: $\Theta(\quad)$

Height: $\log N$
 #node/layer: 2^i
 ($0 \sim \log N$)
 Sum: $1 + 2^1 + 2^2 + \dots + 2^{\log N}$
 $= 2^{\log N} \times 2 - 1 = 2^{N-1} \sim N$

```

1 public static void p4(int N) { m = (int) (17 × 0.4 × 32) = (int) 217.6 = 217
2     int m = (int)((15 + Math.round(3.2 / 2)) *
3         (Math.floor(10 / 5.5) / 2.5) * Math.pow(2, 5));
4     for (int i = 0; i < m; i++) { m = 217
5         System.out.println("hi");
6     }                                     (1)
7 }

```

P4 answer: $\Theta(\quad)$

```

1 public static void p5(int N) {
2     for (int i = 1; i <= N * N; i *= 2) {
3         for (int j = 0; j < i; j++) {
4             System.out.println("moo");
5         }
6     }
7 }

```

$1 \sim N^2 / 2$
 $1 + 2 + 4 + 8 + \dots +$
 $N^2 / 2 + N^2$
 $= N^2 \times 2 - 1 \sim N^2$
 (N^2)

P5 answer: $\Theta(\quad)$

2 A Wild Hilfinger Appears! (Fall 2017, Final)

a. Given the following function definitions, what is the worst-case runtime for $p(N)$?

Assume h is a boolean function requiring constant time.

Answer: $\Theta(\quad)$

```

1 int p(int M) {
2     return r(0, M);
3 }
4
5 int r(int i, int M) {
6     if (i >= M) return 0;
7     if (s(i) > 0) return i;
8     return r(i + 1, M);
9 }
10
11 int s(int k) {
12     if (k <= 0) return 0;
13     if (h(k)) return k;
14     return s(k - 1);
15 }

```

#call to $r(int, int)$

#call to $s(int)$

$s(i) > 0 \Rightarrow$ if $(i \leq 0)$ return 0;
 if $(h(i))$ return k ;
 else return $s(i - 1)$

recursive for M times

k times of calls to s
 that function means:
 if run k times: return 0
 if $h(k)$ holds true: return k (> 0) \Rightarrow trigger return i in $r(int, int)$

Consider worst case: $h(k)$ never holds true, so in every $r(int, int)$, $s(int)$ will be called for i times.

Cost model: #call to $s(int)$; Runtime: $1 + 2 + 3 + 4 + \dots + M \sim M^2 = (N^2)$

b. What is the worst-case runtime for the call $p(N)$? Assume that calls to h require constant time.

Answer: $\Theta(\quad)$

```

1 void p(int M) {           L is like count
2     int L, U;
3     for (L = U = 0; U < M; L += 1, U += 2) {   U: 0 ~ M / 2
4         for (int i = L; i < U; i += 1) {
5             h(i);           i: L ~ U (U-L)
6         }
7     }
8 }

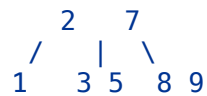
```

L:	0	1	2	3	4	5	6	7	8	9
U:	0	2	4	6	8	10	12	14	16	18
U-L:	0	1	2	3	4	5	6	7	8	9

$\text{sum} \sim (M/2)(M/2-1)/2 = M^2/8 - M/4$
 $\sim M^2 = (N^2)$

3 Tree Time (Spring 2018, Midterm 2)

a. Draw the 2-3 tree that results from inserting 1, 2, 3, 7, 8, 9, 5 in that order.



b. Draw a valid BST of minimum height containing the keys 1, 2, 3, 7, 8, 9, 5.

1, 2, 3, 5, 7, 8, 9

