Greedy algorithm

Minimum Spanning Tree (MST)
Prim's algorithm

Big O

Notation	Time	$n = 100$	$n = 200$
$O(1)$	constant	1	1
$O(\log n)$	log time (base 2)	7	8
$O(n)$	linear	100	200
$O(n \log n)$		700	1600
$O(n^2)$		10000	40000
$O(2^n)$		huge (10 commas)	enormous
$O(n!)$		(40 commas)	super enormous

Towers of Hanoi

```
def towers(n, a, b, c):
    if n == 1:
        print(”Move disk from %s to %s”%(a,b))
    else:
        towers(n-1, a, c, b)
        towers(1, a, b, c)
        towers(n-1, c, b, a)
```

Recurrence equation:
Time(1): 1 move
Time(n): Time($n - 1$) + Time(1) + Time($n - 1$)
Closed form: $O(2^n) - 1$