For
a = subproblems
b = reduction factor
d = pre/post complexity

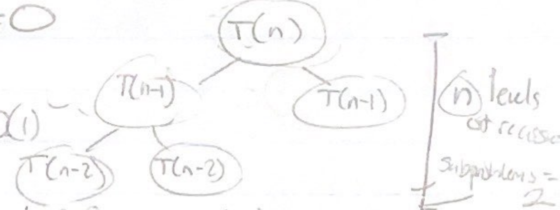
Homework 4: 2.4, 2.5(a-e), 2.17

2.4) Algorithm A: $a=5$ $b=2$ $d=1$

$$\frac{a}{b^d} = \frac{5}{2^1} = \frac{5}{2} > 1 \rightarrow T(n) = O(n^{\log_b a}) = O(n^{\log_2 5})$$
$$O(n^{2.32})$$

Algorithm B: $a=2$ $\frac{n}{b} = n-1$ $d=0$

$$T(n) = 2T(n-1) + O(n^0) = 2T(n-1) + O(1)$$



Runtime = $O(2^n)$ at level k of recursion

Algorithm C: $a=9$ $b=3$ $d=2$

$$\frac{a}{b^d} = \frac{9}{3^2} = \frac{9}{9} = 1 \rightarrow T(n) = O(n^d \log n) = O(n^2 \log n)$$

When comparing the three runtimes, $O(n^2 \log n) \leq O(n^{2.32}) \leq O(2^n)$ so we would want Algorithm C.

2.5) a) $T(n) = 2T(n/3) + 1 \rightarrow a=2$ $b=3$ $d=0$

$\frac{a}{b^d}$

$$\frac{2}{3^0} = 2 > 1 \rightarrow T(n) = O(n^{\log_b a}) = O(n^{\log_3 2}) \boxed{O(n^{.631})}$$

b) $T(n) = 5T(n/4) + n \rightarrow a=5$ $b=4$ $d=1$

$$\frac{5}{4^1} = \frac{5}{4} > 1 \rightarrow O(n^{\log_4 5}) = \boxed{O(n^{1.16})}$$

Homework 4 (cont)

c) $T(n) = 7T(n/7) + n \rightarrow a = 2 \quad b = 7 \quad d = 1$

$$\frac{7}{7^1} = 1 = 1 \rightarrow T(n) = O(n^d \log n) = \boxed{O(n \log n)}$$

d) $T(n) = 9T(n/3) + n^2 \rightarrow a = 9 \quad b = 3 \quad d = 2$

$$\frac{9}{3^2} = \frac{9}{9} = 1 \rightarrow T(n) = O(n^d \log n) = \boxed{O(n^2 \log n)}$$

e) $T(n) = 8T(n/2) + n^3 \rightarrow a = 8 \quad b = 2 \quad d = 3$

$$\frac{8}{2^3} = \frac{8}{8} = 1 \rightarrow T(n) = O(n^d \log n) = \boxed{O(n^3 \log n)}$$

1 sub problem   size in half   constant pre/post time

2.17) To get $O(\log n)$ time $\rightarrow a = 1 \quad b = 2 \quad d = 0$ or $O(1)$

$$\frac{1}{2^0} = \frac{1}{1} = 1 \rightarrow T(n) = O(n^d \log n) = O(1 \log n) = O(\log n)$$

Its binary search

```
SortSearch(A, left, right)
    if left > right
        return False
    i = (left + right) / 2
    if A[i] == i
        return True
    elseif A[i] > i
        SortSearch(A, left, i-1)
    else
        SortSearch(A, i+1, right)
```

// 1 sub problem, size reduced by half, constant time besides recursion