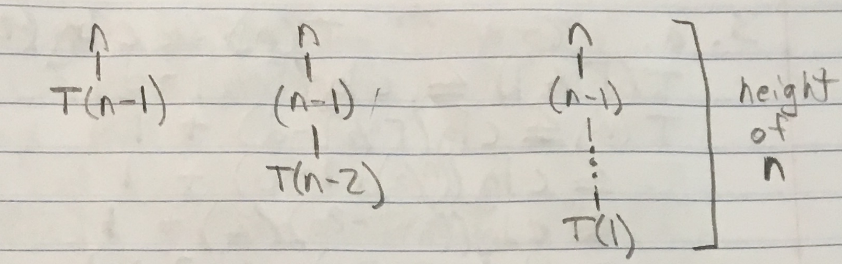
Jake Schiito

Homework 3

1. T(n)



$T(n) = \sum_{k=1}^{n} k = \frac{1}{2}n(n+1) = \frac{1}{2}n^2 + \frac{1}{2}n \rightarrow O(n^2)$

For the master theorem to apply, T(n) must follow the form $T(n) = aT(n/b) + f(n)$ where $a \geq 1$ and $b > 1$. In this example, $T(n-1)$ is not in the form $aT(n/b)$

2. a. $a=3$ $b=2$ $f(n) = n$

$\log_2 3 - \varepsilon = 1$

$O(n^{\log_2 3 - \varepsilon})$ $\varepsilon = \log_2 3 - 1 > 0$

$T(n) = \Theta(n^{\log_2 3})$

b. $a=3$ $b=2$ $f(n) = n^2$

$\log_2 3 + \varepsilon = 2$

$\Omega(n^{\log_2 3 + \varepsilon})$ $\varepsilon = 2 - \log_2 3 > 0$

$3\left(\frac{n}{2}\right)^2 \leq cn^2$ $\frac{3}{4}n^2 \leq cn^2$ for $1 > c \geq \frac{3}{4}$

$T(n) = \Theta(n^2)$

c. $a = 8$ $b = 2$ $f(n) = n^3$

$\log_2 8 = 3$

$n^{\log_2 8} = n^3$

$T(n) = \Theta(n^3 \lg n)$

3. a. Guess: $T(n) \leq c\lg(n-d)$

$T(\lceil n/2 \rceil) \leq c\lg(\lceil n/2 \rceil - d)$

$T(n) \leq c\lg(\lceil n/2 \rceil - d) + 1$

$\leq c\lg((n/2) + 1 - d) + 1$

$= c\lg((n+2-2d)/2) + 1$

$= c\lg(n+2-2d) - c\lg 2 + 1$

$\leq c\lg(n+2-2d)$, $c \geq 1$

$\leq c\lg(n-d)$, $d \geq 2$

Base: $n_0 = 4 \leq n \to \infty$

$T(4) \leq c\lg(4-2)$ where $d = 2$

$c \geq T(4) = T(2) + 1 = T(1) + 2 = 3$

$T(5) \leq 3\lg 3$

$T(5) = T(3) + 1 = T(2) + 2 = T(1) + 3 = 4 \leq 3\lg 3$

$T(n) = O(\lg n)$

b. $T(n) = 2T(\sqrt{n}) + 1$ $m = \lg n$

$T(2^m) = 2T(2^{m/2}) + 1$

Let $S(m) = T(2^m) = 2S(m/2) + 1$

Guess $S(m) \leq cm - d$

$S(m/2) \leq c(m/2) - d$

$S(m) \leq 2c(m/2) - 2d + 1$

$S(m) \leq cm - d$, $d \geq 1$

$T(n) \leq c\lg n - d$

Where $d \geq 1$ and $c \geq T(2) + d$

$T(n) = O(\lg n)$

4. a. FIND_K_SMALLEST (A, n, k)

```
BUILD_MIN_HEAP(A)
array B[1...k]
for i=1 to k do
    B[i] = HEAP_EXTRACT_MIN(A)
return B
```

b. $T(n) \leq O(n) + O(1) + k(O(1) + O(\lg n)) + O(1)$

$T(n) \leq O(n) + O(k) + O(k \lg n) = O(n) + O(k \lg n)$

$k = \lg n$

$T(n) \leq O(n) + O(\lg n) + O(\lg n \lg n)$

$$\lim_{n\to\infty} \frac{n}{(\lg n)^2} = \lim_{n\to\infty} \frac{1}{2(1/n)\lg n} = \lim_{n\to\infty} \frac{n}{2\lg n} = \lim_{n\to\infty} \frac{1}{2/n} = \infty$$

$n > (\lg n)^2 > \lg n$ as $n \to \infty$

$T(n) = O(n)$

Therefore, assuming $k = \lg n$, the runtime of BUILD_MIN_HEAP dominates the program and becomes the bound of $O(n)$ which is less than SORT-THEN-SELECT-K of $O(n \lg n)$

c. If $k = 1$

SORT-THEN-SELECT-K: $T(n) = O(n \lg n + 1) = O(n \lg n)$

FIND_K_SMALLEST: $T(n) = O(n) + O(\lg n) = O(n)$

FIND-MINIMUM: $T(n) = O(n)$

The heap algorithm has the same asymptotic runtime as FIND-MINIMUM when $k = 1$ because Building the heap dominates the runtime and takes the same asymptotic time as one iteration. merge sort takes $O(n \lg n)$ so it will be slower for finding 1 value.