Caleb Faruki
Dr. Xiaoxu Han
CISC4080 Algorithms
November 4, 2013

Homework 2

Part 1A

1

$$T_n = T_{n-1} + c^n, \ \ where \ \ c > 1$$
$$T_2 - T_1 = c^2$$
$$T_3 - T_2 = c^3$$
$$\ldots$$
$$T_n - T_{n-1} = c^n$$
$$T_n - T_1 = c^2 + c^3 + \ldots + c^n$$
$$T_n - T_1 = c^2 \cdot (1 + c + c^2 + c^3 + \ldots + c^{n-1})$$

$$1 + r + r^2 + \ldots + r^{k-1} \ = \ \frac{r^k - 1}{r - 1}$$
$$1 + c + c^2 + \ldots + c^{n-1} \ = \ \frac{c^n - 1}{c - 1}$$

$$T_n - T_1 = c^2 \cdot \frac{c^n - 1}{c - 1}$$

$$T_n - T_1 = \frac{c^{n+1} - c^2}{c - 1} \ = \ \boldsymbol{\Theta(c^n)}$$

2

$$T_n = 16\mathrm{T}\left(\frac{n}{4}\right) + n^3$$
$$a = 16, \ \ b = 4, \ \ d = 3$$

$$a < b^d \rightarrow 16 < 4^3$$

$$\boldsymbol{T_n = \Theta(n)}$$

3

$$T_n = 25\mathrm{T}\left(\frac{n}{36}\right) + n^{3/2} logn$$
$$a = 25, \ \ b = 36, \ \ d = 1$$

$$a < b^d \rightarrow 25 < 36^1$$
$$\boldsymbol{T_n = \Theta(n)}$$

Part 2A

$$T_1 = 1, \ \ T_n = 2\mathrm{T}\left(\frac{n}{2}\right) + c$$

$$n = \frac{n}{2,} \quad so\ldots$$

$$T_{\frac{n}{2}} = 2\mathrm{T}\left(\frac{\left(\frac{n}{2}\right)}{2}\right) + c$$
$$= \ \ 4\mathrm{T}\left(\frac{n}{4}\right) + 2\mathrm{c} + c$$

$$T_n = 2^2 T\left(\frac{n}{2^2}\right) + 2\mathrm{c} + c$$
$$T_{\frac{n}{4}} = 2\mathrm{T}\left(\frac{n}{8}\right) + c$$

$$T_n = 2^2\left[2\mathrm{T}\left(\frac{n}{8}\right) + 2\mathrm{c} + c\right]$$
$$T_n = 2^3 T\left(\frac{n}{8}\right) + 2^2 c + 2\mathrm{c} + c$$

$$= 1 + r + r^2 + \ldots + r^{k-1}$$
$$= \frac{r^{k-1} - 1}{r - 1}$$
$$= \frac{1 + c^2 * c^{n-1} - 1}{c - 1}$$
$$= \frac{1 + c^2 * c^{n-1} - c^2}{c - 1}$$
$$T_n = \frac{1 + c^{n+1} - c^2}{c - 1} = \boldsymbol{\Theta(n)}$$

Part 2B

Worst: $T(n)=\Theta(n)$

Average: $T(n)=\Theta(n)$

Part 3B

Best, worst, and average case performance for isSorted() is linear. Therefore, time complexity for quickSort() in any case is simply $T(n)=\Theta(quickSort() + n)$. Every call to quickSort equals one call to isSorted().

$T(n)=\Theta(n)+2\mathrm{T}(\frac{n}{2})$ isSorted is subsumed by Θ(n), since it's contribution to the workload is additive, not multiplicative. Θ(n) is still equal to n.

Best: $T(n)=\Theta(nlog(n))$

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

Average: $T(n)=\Theta(nlog(n))$

Part 2C

Best: $T(a,b)=\Theta(1)$

Worst: $T(a,b)=\Theta(\log(b))$ Not sure how to calculate average-case.