CSCI 432 - Assignment G4

Sean White, Kierstyn Brandt, Rostik Mertz, Norman Tang

October 17, 2017

Question A: Solve the following recurrences:

a) Show that $T(n) = T(n-1) + n$ is $O(n^2)$ using the substitution method.

To say that $T(n) = T(n-1) + n$ is $O(n^2)$ is to say that $T(n)$ has an asymptotic upper bound of n^2. To show this is true we need to make a guess at a closed upper bound, so let that be $cn^2 - b$ where c and b are arbitrary constants. Now our relation is $T(n) \leq cn^2 - b$ so we see

$$
\begin{aligned}
T(n) &= T(n-1) + n \\
&\leq c(n-1)^2 - b + (cn^2 - b) \\
&= c(n^2 - 2n + 1) + cn^2 - 2b \\
&= 2cn^2 - 2cn - 2b + c \\
&= cn^2 - cn - 2b + c \\
&\leq cn^2 - b
\end{aligned}
$$

This is true $\forall c \geq b \geq 0$ so we set $c = 1$ and $b = 0$ which is allowed by the definition of big O and get that $T(n) \leq O(n^2)$.

b) Use a recursion tree to determine a good asymptotic upper bound for $T(n) = T(n/2) + n^2$.

The height of the tree is $\log(n)$ and at each step (of which there are $\log(n)$ of) we do n^2 steps, resulting in an upper bound of $O(n^2 log(n))$

c) Use the master method to solve $T(n) = 2T(n/4) + 1$

To start the master method we set the variables $a = 2, b = 4 and f(n) = 1$ where $a \geq 1$, $b > 1$ and $f(n)$ is asymptotically positive. Next we calculate

$n^{log_b a} = n^{log_4 2} = n^{1/2}$. This then puts us into case 1 of master's theorem. We can then use the identity function $f(n) = O(n^{log_b a - \varepsilon}$ for some $\varepsilon > 0$ We then plug in the values $f(n) = O(n^{log_4 3 - \varepsilon})$ where $\varepsilon = 1/2$. This gives us a value of 1 which is equal to $f(n)$. Therefore $T(n) = \Omega(n^{log_b a})$ which equals $T(n) = \Omega(n^{log_4 2}) = \Omega(n^{1/2})$

d) Use the master method to solve $T(n) = 2T(n/4) + \sqrt{n}$

$T(n) = 2T(\frac{n}{4}) + \sqrt{n}$, can otherwise be written as: $T(n) = 2T(\frac{n}{4}) + n^{1/2}$. We set our variables as: $a = 2, b = 4, f(n) = n^{1/2}$ and $\frac{1}{2}$, s.t. $a \geq 1 \& b > 1$, and $f(n)$ is an asymptotically positive function on n. $log_4 2 = \frac{1}{2} = c$, so this is a case 2 of the theorem. We then use the identity function: $f(n) = \theta(n^{log_b a} log^{k+1} n)$. We plug in the values and get: $\sqrt{n} log n$. Hence, $T(n) = \theta\sqrt{n} log^1 n$

e) Use the master method to solve $T(n) = 2T(n/4) + n$

As seen before, the master method is a useful tool when solving recurrences in the form $T(n) = aT(\frac{n}{b}) + f(n)$ where a and b are constants such that $a \geq 1$ and $b > 1$, and $f(n)$ is an asymptotically positive function on n. Given the recurrence $T(n) = 2T(n/4) + n$ we see that $a = 2$, $b = 4$, and $f(n)$ is the identity function. Since the identity function $f(n) = n$ is $\Omega(n)$ and $\Omega(n) > \Omega(n^{log_4 2})$ we are in the case where $f(n)$ is $\Omega(n^{log_b a + \varepsilon})$, so we are in the case where $f(n)$ grows faster than the leaves of the recursion tree. Therefore, by the master theorem, $T(n)$ is $\Theta(f(n))$ so $T(n) = \Theta(n)$.

Question B: Consider the randomized algorithm to compute the smallest enclosing ball. Consider the example set in the figure included on the assignment sheet. Suppose that the 'random' choice of vertex always chooses such that the vertex has the smallest index. What are the return values of each recursive call?