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 $T(n) = 2T(n/4) + 1$

$n^{log_b a} = n^{\log_4 2} = n^{1/2}$ f(n) = $\Omega(n^{\log_4 3+}) where = 1/2$ f(n) = $\Omega(n^{1/2+1/2})$ 2f(n/4) $\leq$ cf(n) if c = 1/2 Then T(n) = $\Theta(1)$ d) Use the master method to solve $T(n) = 2T(n/4) + \sqrt{n}$

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

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?