TA: Jade Cheng
ICS 311
Quiz Solution #5
March 11, 2009

Quiz #5:

What is the average time complexity (Θ) of each of the following actions? Briefly justify your answers. [5 pts]

a. Insert an item into a Binary Search Tree.

Answer:

$\Theta(n)$. Assuming the BST is not balanced

$\Theta(lg\, n)$. Assuming the tree is balanced or randomly built

b. Remove an item from a Red-Black Tree .

Answer:

$\Theta(lg\, n)$. Red-Black Tree guarantee $\Theta(lg\, n)$ time per access by adjusting tree structure after every modification. The height of an R-B tree cannot be any shorter than $lg(n+1)$ but also no taller than $2 \times lg(n+1)$. This is obviously logarithmic, so a red black tree guarantees approximately the best case for a binary search tree, which is $\Theta(lg\, n)$.

c. Insert an item into a hash table that uses lists for collision management.

Answer:

$\Theta(1)$ Collision resolving using linked list has a insertion access time $\Theta(1)$ by inserting the elements in the front side of the linked lists

d. Quicksort Q times.

Answer:

$\Theta(Q\,lg\,Q)$. Quicksort is proven to have an average run time expectation $\Theta(n\,lg\,n)$, where n is the input size. Therefore, for an input size of Q, it takes $\Theta(Q\,lg\,Q)$.

e. Add K items to a heap.

Answer:

$\Theta(K\,lg\,n)$. Heap adding and deleting operations takes $\Theta(lg\,n)$. For small K, we omit the lower order term K, the overall time complexity of adding K items is $\Theta(K\,lg\,n) \approx \Theta(lg\,n)$. For large K that is proportional to the heap size, the overall time complexity is $\Theta(K\,lg\,n) \approx \Theta(n\,lg\,n)$.

An algorithm is described by: $T(n)=3n^3+2n^2+5n\,lg\,n+7$. What is the time complexity (Θ) of this algorithm? Prove your answer. [4 pts]

Answer:

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

Goal: there exists positive constants c_1, c_2, and n_0, such that $0 \le c_1 g(n) \le T(n) \le c_2 g(n)$, where $g(n)=n^3$, holds for all $n \ge n_0$.

Let's pick $c_1=2$, $c_2=4$, and $n_0=8$.

<u>Prove the upper bound $T(n) \le c_2 g(n)$ by induction:</u>

For simplification purpose, we will first prove $T(n)$ is upper bounded by $g'(n)=3n^3+7n^2+7$, then prove $g'(n)=3n^3+7n^2+7$ is upper bounded by $c_2 g(n)$. Based on the definition of Θ notation, if $T(n) \le g'(n) \le c_2 g(n)$ holds for all $n \ge n_0$, we prove $T(n) \le c_2 g(n)$ holds for all $n \ge n_0$.

Step 1: Prove $T(n) \le g'(n)$

To be proved: $3n^3+2n^2+5n\,lg\,n+7 \le 3n^3+7n^2+7 \rightarrow lg\,n \le n$

Base case: $lg\,8 = 3 \le 8$. Base case holds.

Assume: $lg(n-1) \le n-1$ holds.

$$
\begin{aligned}
lg(n-1) &= lg\frac{n(n-1)}{n} \\
&= lg\,n + lg\frac{n-1}{n} \\
&\le n-1
\end{aligned}
$$

Therefore:

$$
\begin{aligned}
lg\,n &\le n-1-lg\frac{n-1}{n} \\
&= n-(1+lg\frac{n-1}{n}) \\
&= n-lg\frac{2(n-1)}{n} \\
&= n-lg(1+\frac{n-2}{n})
\end{aligned}
$$

$T(n) \le g'(n)$ holds, if inequation $T(n-1) \le g'(n-1)$ and $lg(1+(n-2)/n) \ge 0$ holds for all $n \ge n_0$.

$T(n-1) \le g'(n-1)$ holds for $n \ge n_0$ is the assumption, so we just need to prove $lg(1+(n-2)/n) \ge 0$ holds for all $n \ge n_0$.

$lg(1+(n-2)/n) \ge 0$ holds if $(n-2)/n \ge 0$, which obvious does for all $n \ge n_0 = 8$

So far, we proved $T(n) \le g'(n)$ holds for all $n \ge n_0$.

Step 2: Prove $g'(n) \le c_2 g(n)$

Base case: $3\times 8^3+2\times 8^2+5\times 8\times lg\,8+7=1791$

$c_2 g(8) = 4\times 8^3 = 4\times 512 = 2048$

Base case holds

Assume: $g'(n-1) \leq c_2 g(n-1)$ holds

$$
\begin{aligned}
g'(n-1) &= 3(n-1)^3 + 7(n-1)^2 + 7 \\
&= (3n^3 + 7n^2 + 7) - 9n^2 + 7n + 4 \\
&= g'(n) - 9n^2 + 7n + 4 \\
&\leq c_2 g(n-1) \\
&= 4(n-1)^3 \\
&= 4(n^3 - 3n^2 + 3n - 1) \\
&= c_2 g(n) - 12n^2 + 12n - 4
\end{aligned}
$$

Therefore:

$$
\begin{aligned}
g'(n) &\leq c_{2g}(n) - 12n^2 + 12n - 4 + 9n^2 - 7n - 4 \\
&= c_{2g}(n) - 3n^2 + 5n - 8 \\
&= c_{2g}(n) - (3n^2 - 5n + 8)
\end{aligned}
$$

$g'(n) \leq c_2 g(n)$ holds for all $n \geq n_0$, if $g'(n-1) \leq c_2 g(n-1)$ and $3n^2 - 5n + 8 \geq 0$ holds for all $n \geq n_0$.

$g'(n-1) \leq c_2 g(n-1)$ holds for $n \geq n_0$ is the assumption, and it is obvious that $3n^2 - 5n + 8 \geq 0$ holds for all $n \geq n_0 = 8$.

So, we proved $g'(n) \leq c_2 g(n)$ holds for all $n \geq n_0$.

Combine Step 1 and Step 2, we proved $T(n) \leq g'(n)$ and $g'(n) \leq c_2 g(n)$, where $g'(n) = 3n^3 + 7n^2 + 7$, hold for all $n \geq n_0$. Therefore, $T(n) \leq c_2 g(n)$ holds for all $n \geq n_0$.

<u>Prove the lower bound $0 \leq c_1 g(n) \leq T(n)$ by simple math:</u>

It is obvious $0 \leq c_1 g(n) = 2n^3$ holds for all $n \geq n_0 = 8$ and $c_1 g(n) \leq T(n)$, which can be written as $2n^3 \leq 3n^3 + 2n^2 + 5n \lg n + 7$, holds for all $n \geq n_0 = 8$.

As we proved both the upper bound and lower bound, we proved that $0 \leq c_1 g(n) \leq T(n) \leq c_2 g(n)$, where $g(n) = n^3$, $c_1 = 2$, $c_2 = 4$, and $n_0 = 8$, holds for all $n \geq n_0$. Therefor, the objective function $T(n)$ is bounded by $g(n) = n^3$, $T(n) = \Theta(n^3)$.

Circle all sort algorithms below that are stable. [2 pts]

Answer:

Bucket Sort	→	stable
Counting Sort	→	stable
Insertion Sort	→	stable
Merge Sort	→	stable
Quicksort	→	unstable
Radix Sort	→	stable
Selection Sort	→	stable
Shell Sort	→	unstable