CS 240E S25

Tutorial 1

May 9

Q1

Show that $n \in \omega\left(2^{\sqrt{\lg n}}\right)$ using the definition.

Which is, show that $\forall c > 0, \exists n_0 \geq 0$ such that $n \geq c \cdot \left|2^{\sqrt{\lg n}}\right|$ for all $n \geq n_0$.

Q2

Let $f(x)$ be a positive monotone function with domain R^+. Assume that $f(x) \le x$ for all even integers x. Show that $f(x) \in O(x)$.

Q3

Let $f(n), g(n)$ be eventually positive. In lecture we saw that if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, then $f(n) \in o(g(n))$. Prove the converse.

Q4

Give a tight bound on the runtime of the following algorithm as a function of n.

```
k = 1
for(i = 1; i <= n; ++i):
    j = 0
    while j <= n:
        j += k
    k *= 2
```

Additional problems

Q5

Give an exact bound on the n-th Harmonic number $H_n = \sum_{k=1}^{n} \frac{1}{k}$.

Note: this question requires bounds with integration.

Q6

Assuming any appropriate base case, give asymptotic upper and lower bounds (make them as tight as you can) for $T(n)$ if,

a) $T(n) = 3T\left(\frac{n}{3}\right) + \frac{n}{\log_3 n}$
b) $T(n) = T(n-1) + \frac{1}{n}$
c) $T(n) = \sqrt{n}T\left(\sqrt{n}\right) + n$

Enrich content

Master Theorem

Master theorem gives a general recurrence solution for most of the divide-and-conquer algorithms.

Suppose $T(n)$ satisfies $T(n) \leq aT\left(\frac{n}{b}\right) + n^c$ for all sufficiently large n:

if $\log_b a < c$, then $T(n) \in O(n^c)$,

if $\log_b a = c$, then $T(n) \in O(n^c \log n)$, and

if $\log_b a > c$, then $T(n) \in O\left(n^{\log_b a}\right)$.

* This version of Master Theorem is simplified and some details are omitted. The full version will be introduced in upper-level courses.