CS 4720/5720 Design and Analysis of Algorithms

Homework #1

Student: Daniel Frey

Answers to homework problems:

1. Book Problem: 2.2.3

   (a) $(n^2+1)^{10} \approx (n^2)^{10} = n^{20} \in \Theta(n^{20})$

   $\lim_{n\to\infty} \frac{(n^2+1)^{10}}{n^{20}} = \lim_{n\to\infty} \frac{(n^2+1)^{10}}{(n^2)^{10}} = \lim_{n\to\infty}(\frac{(n^2+1)}{n})^{10} = \lim_{n\to\infty}(1+\frac{1}{n^2})^{10} = 1 \in \Theta(n^{20})$

   (b) $\sqrt{10n^2+7n+3} \approx \sqrt{10n^2} = n\sqrt{10} \in \Theta(n)$

   $\lim_{n\to\infty} \frac{\sqrt{10n^2+7n+3}}{n} = \lim_{n\to\infty} \sqrt{\frac{10n^2+7n+3}{n^2}} = \lim_{n\to\infty} \sqrt{\frac{10n^2}{n^2}+\frac{7n}{n^2}+\frac{3}{n^2}} = \lim_{n\to\infty} \sqrt{10+\frac{7}{n}+\frac{3}{n^2}} = \sqrt{10} \in \Theta(n)$

   (c) $2n\log(n+2)^2+(n+2)^2\log\frac{n}{2} = 2n*2\log(n+2)+(n+2)^2(\log n-\log 2) = [max(\Theta(n\log n),\Theta(n^2\log n))] \in \Theta(n^2\log n)$

   (d) $2^{n+1}+3^{n-1} = 2^n*2+3^n*\frac{1}{3} \in [max(\Theta(2^n),\Theta(3^n))] = \Theta(3^n)$

   (e) $\lfloor\log_2 n\rfloor \approx \log_2 n \in \Theta(\log n)$
   $x-1 < \lfloor x\rfloor \le x \Rightarrow \lfloor log_2 n\rfloor \le \log_2 n$
   $\lfloor\log_2 n\rfloor > \log_2 n - 1 \ge \log_2 n - \frac{1}{2}\log_2 n = \frac{1}{2}\log_2 n \in \Theta(\log_2 n) = \Theta(\log n)$

2. Solve the recurrence relations.

   (a) $T(n) = 2T(\frac{n}{2})+n^3 \Rightarrow T(x) = 2T(\frac{x}{2})+x^3$
   Let $x=\frac{n}{2}, T(\frac{n}{2}) = 2T(\frac{n}{4})+\frac{1}{8}n^3$
   $T(n) = 2[2T(\frac{n}{4})+\frac{1}{8}n^3]+n^3 = 4T(\frac{n}{4})+\frac{1}{4}n^3+n^3$
   Let $x=\frac{n}{4}, T(\frac{n}{4}) = 2T(\frac{n}{8})+\frac{1}{64}n^3$
   $T(n) = 4[2T(\frac{n}{8})+\frac{1}{64}n^3]+\frac{5}{4}n^3 = 8T(\frac{n}{8})+\frac{1}{16}n^3+\frac{1}{4}n^3+n^3$
   $T(n) = 2^kT(\frac{n}{2^k})+n^3\sum_{i=0}^{k-1}(\frac{1}{4})^i$
   $\frac{n}{2^k} = 1 \Rightarrow n = 2^k \Rightarrow \log_2 n = k \Rightarrow nT(1)+n^3\sum_{j=0}^{\log_2 n-1}(\frac{1}{4})^j$
   $T(n) = nT(1)+n^3(\frac{(1-\frac{1}{4})^{\log_2 n}}{1-\frac{1}{4}})$
   $\lim_{n\to\infty} nT(1)+n^3(\frac{(1-\frac{1}{4})^{\log_2 n}}{1-\frac{1}{4}}) = \frac{1}{1-\frac{1}{4}} = \frac{4}{3} \Rightarrow nT(1)+n^3(\frac{4}{3}) \in \Theta(n^3)$

   (b) $T(n) = T(\sqrt{n})\sqrt{n}+n \Rightarrow T(x) = T(\sqrt{x})\sqrt{x}+x$
   Let $x=\sqrt{n}, T(\sqrt{n}) = T(n^{\frac{1}{4}})n^{\frac{1}{4}}+n^{\frac{1}{2}}$
   $T(n) = [T(n^{\frac{1}{4}})n^{\frac{1}{4}}+n^{\frac{1}{2}}]\sqrt{n}+n = T(n^{\frac{1}{4}})n^{\frac{3}{4}}+2n$
   Let $x=n^{\frac{1}{4}}, T(n^{\frac{1}{4}}) = T(n^{\frac{1}{8}})n^{\frac{1}{8}}+n^{\frac{1}{4}}$
   $T(n) = [T(n^{\frac{1}{8}})n^{\frac{1}{8}}+n^{\frac{1}{4}}]n^{\frac{3}{4}}+2n = T(n^{\frac{1}{8}})n^{\frac{7}{8}}+3n$
   $T(n) = T(n^{\frac{1}{2^k}})n^{\frac{2^k-1}{2^k}}+kn \Rightarrow \frac{2^k-1}{2^k} = 1-2^{-k}$
   $n^{\frac{2^k-1}{2^k}} = 2 \Rightarrow (n^{\frac{2^k-1}{2^k}})^{2^k} = (2)^{2^k} \Rightarrow n = 2^{2^k} \Rightarrow \log_2\log_2 n = k$
   $T(n) = T(2)\frac{n}{2}+n\log\log n \in \Theta(n\log\log n)$

   (c) $T(n) = 2T(\frac{n}{2})+\frac{n}{\log n} \Rightarrow T(x) = 2T(\frac{x}{2})+\frac{x}{\log x}$
   Let $x=\frac{n}{2}, T(\frac{n}{2}) = 2T(\frac{n}{4})+\frac{\frac{n}{2}}{\log\frac{n}{2}} = 2T(\frac{n}{4})+\frac{\frac{n}{2}}{\log n-1}$

$T(n) = 2[2T(\frac{n}{4}) + \frac{\frac{n}{2}}{\log \frac{n}{2}}] + \frac{n}{\log n} = 4T(\frac{n}{4}) + \frac{n}{\log n-1} + \frac{n}{\log n}$
Let $x = \frac{n}{4}, T(\frac{n}{8}) = 2T(\frac{n}{8}) + \frac{\frac{n}{4}}{\log \frac{n}{4}} = 2T(\frac{n}{8}) + \frac{\frac{n}{4}}{\log n-2}$
$T(n) = 4[2T(\frac{n}{8}) + \frac{\frac{n}{4}}{\log \frac{n}{4}}] + \frac{n}{\log n-1} + \frac{n}{\log n} = 8T(\frac{n}{4}) + \frac{n}{\log n-2} + \frac{n}{\log n-1} + \frac{n}{\log n}$
$T(n) = 2^k T(\frac{n}{2^k}) + \sum_{i=0}^{k-1} \frac{n}{\log n-i}$
$\frac{n}{2^k} = 1 \Rightarrow n = 2^k \Rightarrow \log_2 n = k$
$T(n) = 2^{\log_2 n} T(\frac{n}{n}) + \sum_{i=0}^{\log_2 n-1} \frac{n}{\log_2 n-i} \Rightarrow j = \log_2 n - i$
$T(n) = nd + \sum_{j=0}^{\log_2 n-1} \frac{n}{j} \Rightarrow nH \log_2 n \Rightarrow n \log\log n \in \Theta(n \log\log n)$

(d) Using Master Thm: Case 1: $a = 3, b = 2, f(n) = nlogn$
$n^{\log_2(a-\epsilon)} = n^{\log_2 3-\epsilon} = n^{1.58} \Rightarrow T(n) = \Theta(n^{1.58})$

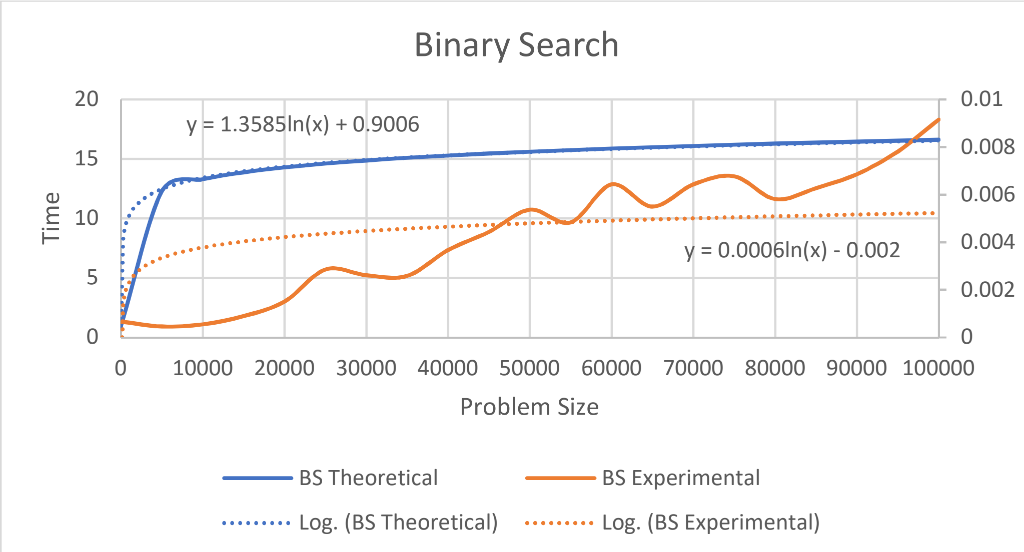
3. BinarySearch(A[0, n-1], key, lower, upper)
if upper >= lower and A size > 0
mid = lower + (upper - lower) /2
if key == A[mid], return mid
else if key < A[mid] return BinarySearch(A, key, lower, mid-1)
else return BinarySearch(A, key, mid+1, upper)
else return -1

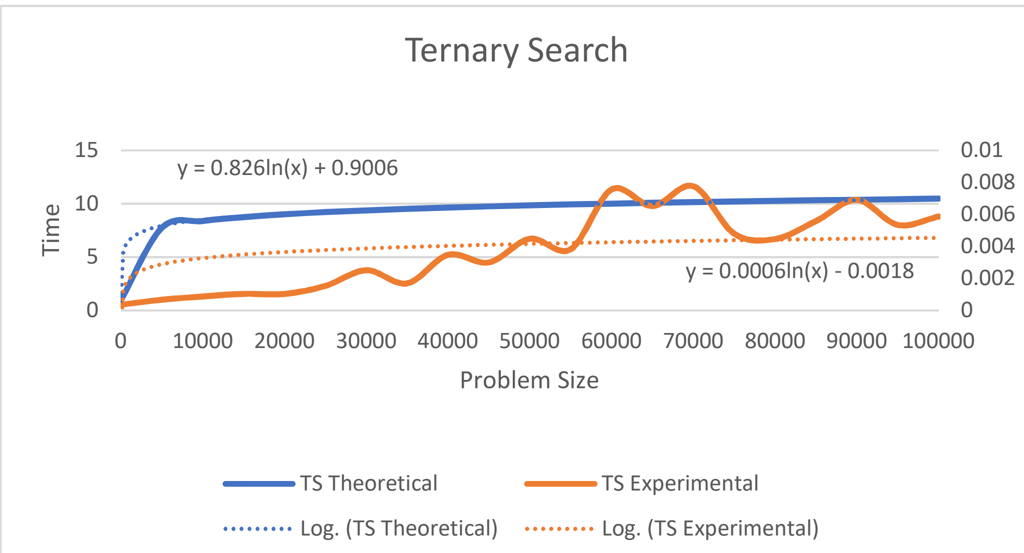
$T(n) = T(\frac{n}{2}) + 1, T(x) = T(\frac{x}{2}) + 1$
Let $x = \frac{n}{2}, T(\frac{n}{2}) = T(\frac{n}{4}) + 1$
$T(n) = T(\frac{n}{4}) + 1 + 1$
Let $x = \frac{n}{4}, T(\frac{n}{4}) = T(\frac{n}{8}) + 1$
$T(n) = T(\frac{n}{8}) + 1 + 1 + 1$
$T(n) = T(\frac{n}{2^k}) + k$
$\frac{n}{2^k} = 1 \Rightarrow n = 2^k \Rightarrow \log_2 n = k$
$T(n) = T(\frac{n}{n}) + \log_2 n = T(1) + \log_2 n = 1 + \log_2 n \in \Theta(\log n)$

TernarySearch(A[0, n-1], key, lower, upper)
if upper >= lower and A size > 0
Lmid = lower + (upper - lower) /3
Umid = Lmid + (upper - lower) /3
if key == A[Lmid], return Lmid
else if key == A[Rmid], return Umid
else if key < A[Lmid] return TernarySearch(A, key, lower, Lmid-1)
else if key > A[Umid] return TernarySearch(A, key, Umid+1, upper)
else return TernarySearch(A, key, Lmid+1, Umid-1)
else return -1

$T(n) = T(\frac{n}{3}) + 1, T(x) = T(\frac{x}{3}) + 1$
Let $x = \frac{n}{3}, T(\frac{n}{3}) = T(\frac{n}{9}) + 1$
$T(n) = T(\frac{n}{9}) + 1 + 1$
Let $x = \frac{n}{9}, T(\frac{n}{9}) = T(\frac{n}{27}) + 1$
$T(n) = T(\frac{n}{27}) + 1 + 1 + 1$
$T(n) = T(\frac{n}{3^k}) + k$
$\frac{n}{3^k} = 1 \Rightarrow n = 3^k \Rightarrow \log_3 n = k$
$T(n) = T(\frac{n}{n}) + \log_3 n = T(1) + \log_3 n = 1 + \log_3 n \in \Theta(\log n)$

The graphs below both follow somewhat similar data trends. The experimental results might be slightly different because of the random numbers that had been generated for the problem.


Binary Search
y = 1.3585ln(x) + 0.9006
y = 0.0006ln(x) - 0.002
Time
Problem Size
20
15
10
5
0
0.01
0.008
0.006
0.004
0.002
0
0 10000 20000 30000 40000 50000 60000 70000 80000 90000 100000
BS Theoretical
BS Experimental
Log. (BS Theoretical)
Log. (BS Experimental)


Ternary Search
y = 0.826ln(x) + 0.9006
y = 0.0006ln(x) - 0.0018
Time
Problem Size
15
10
5
0
0.01
0.008
0.006
0.004
0.002
0
0 10000 20000 30000 40000 50000 60000 70000 80000 90000 100000
TS Theoretical
TS Experimental
Log. (TS Theoretical)
Log. (TS Experimental)