**COP 3530**
**Section U03**
**Spring 2017**

# MIDTERM EXAM ANSWERS

**Question 1** (15 points)

The program below uses the divide and conquer method to minimize the number of multiplications needed to compute x to the power n. The standard approach is to compute $x^n = x * x \ldots * x$, where the multiplication $*$ is performed $n-1$ times.

```
// assume that x != 0 and n >= 0
public static double xTon(double x, int n)
{
    // base cases
    if (n == 0)
        return 1.0;
    if (n == 1)
        return x;
    int half = n / 2; // get the half
    double y = xToN(x,half); // get x to half of x
    if (n % 2 == 0)
        return y * y ;
    else // n is odd
        return y * y * x;
}
```

Let $T(n)$ be the largest number of multiplications performed by this method. Write the boundary value $T(1)$ and the recurrence relation for $T(n)$. Then solve the recurrence relation.

**Solution:**

$T(1) = 0$

$T(n) = T(n/2) + 1$ if $n$ is even

$T(n) = T(n/2) + 2 \qquad$ if $n$ is odd

The Master Theorem says that the solution to $T(n) = aT(n/2) + n^c$ is $T(n) = \Theta(n^c logn)$ when $a = b^c$. In our case $a = 1, b = 2, c = 0$, so $a = b^c$.

The constant of in front of $n^c$ is not relevant, so we get the same solution for $T(n) = T(n/2) + 2$.

In short, $T(n) = \Theta(logn)$.

**Grading Criteria:**

1. 3 points for $T(1)$
2. 6 points for $T(n)$
3. 6 points for $\Theta$

**Question 2.** (20 points)

The recurrence $T(n)$ is defined as follows:

$T(n) = c \qquad$ if $n \leq 4$

$T(n) = T(n/2) + T(n/3) + T(n/4) + n^2 \qquad$ if $n > 4$

Find $f(n)$ such that $T(n) = \Theta(f(n))$.

**Solution 1:** The easiest way is to see that $T(n)$ increases in $n$. So, we have (1).

(1) $3T(n/4) \leq T(n) \leq 3T(n/2)$

Now let us define the recurrences $T_1$ and $T_2$ below.

$T_1(n) = c \qquad$ if $n \leq 4$

$T_1(n) = 3T_1(n/4) \qquad$ if $n > 4$

$T_2(n) = c \qquad$ if $n \leq 4$

$T_2(n) = 3T_2(n/2) \qquad$ if $n > 4$

One can use (1) to show (2).

(2) $T_1(n) \leq T(n) \leq T_2(n)$.

We apply the Master Theorem to both $T_1$ and $T_2$ and get the same $\Theta$, $T_1(n) = \Theta n^2$, $T_2(n) = \Theta n^2$. So, $T(n) = \Theta(n^2)$.

**Solution 2:** One could make an educated guess that $T(n) = \Theta(n^2)$ and prove it by induction. We need to show that $T(n) = O(n^2)$ and $T(n) = \Omega(n^2)$.

Here is the proof for $T(n) = O(n^2)$. Assume that (3) holds for all $m \leq n$ and we need to show (4).

(3) $T(m) \leq c_1 m^2$

(4) $T(n+1) \leq c_1 (n+1)^2$

Now let's show (4).

$T(n) = T((n+1)/2) + T((n+1)/3) + T((n+1)/4) + (n+1)^2$ definition of $T(n+1)$

$\leq c_1 \lfloor (n+1)/4 \rfloor^2 + c_1 \lfloor (n+1)/3 \rfloor^2 + c_1 \lfloor (n+1)/2 \rfloor^2 + (n+1)^2$ by (3)

$\leq c_1 ((n+1)/4)^2 + c_1 ((n+1)/3)^2 + c_1 ((n+1)/2)^2 + (n+1)^2$ because $\lfloor x \rfloor \leq x$

$= c_1 (\frac{1}{16} + \frac{1}{9} + \frac{1}{4})(n+1)^2 + (n+1)^2$

$= (c_1 \frac{61}{144} + 1)(n+1)^2$

$\leq c_1 (n+1)^2$ for $c_1 \geq \frac{144}{83}$.

The proof for $T(n) = \Omega(n^2)$ is trivial because $T(n) \geq (n+1)^2$ for all $n > 4$.

**Grading Criteria:** 1. If you choose the second approach you loose 3 points.

2. Skipping steps : -2 points for each missing step.

3. -5 points for a bad presentation (extra junk, missing steps, lacking direction, etc).

4. 6 points for identifying the $\Theta$.

**Question 3.**(25 points)

Solve the recurrence relation

$T(0) = 0$

$T(n) = (1/n)(T(0) + T(1) + \ldots + T(n-1)) + cn$ if $n > 0$

Here $c$ is a positive constant.

**Solution:** For $n > 1$ we have (1) and (2).

(1) $nT(n) = T(0) + T(1) + \ldots + T(n-2) + T(n-1) + cn^2$

(2) $(n-1)T(n-1) = T(0) + T(1) + \ldots + T(n-2) + c(n-1)^2$

We subtract (2) from (1) and get (3).

(3) $nT(n) - (n-1)T(n-1) = T(n-1) + 2cn - c$

(3) reduces to (4).

(4) $T(n) = T(n-1) + 2c - c\frac{1}{n}$

The last formula produces, by induction, (5).

(5) $T(n) = 2cn - c(1 + \frac{1}{1} + \ldots + \frac{1}{n})$

These function is approximated by $f(n) = 2cn - cln(n+1)$, the difference being less than c.

If we want to express it in terms of the logarithm in base 2 we get (6).

(6) $f(n) = 2cn - clog(n+1)/log(e)$.

**Grading Criteria:**

1. 12 points for getting (3)
2. 15 points for getting (4)
3. 21 points for (5)
4. 25 points for $f$
5. 5 points for trying

**Question 4.** (35 points)

Prove Theorem 10.7, pp 450, from Mark Weiss'e book which says,

The solution to the equation $T(n) = aT(n/b) + \Theta(n^k log^p n)$ where $a \geq 1, b > 1, p \geq 0$ is

$T(n) = O(n^{log_b a})$ if $a > b^k$

$T(n) = O(n^k log^{p+1}(n))$ if $a = b^k$

$T(n) = O(n^k log^p(n))$ if $a < b^k$

This a generalization of the Master Theorem.

**Proof:**

**Part 4.1:** We will prove the theorem for $n = b^m$.

Since $T(n) = aT(n/b) + \Theta(n^k log^p n)$, we know that there is a constant $c$ such that (1) holds.

(1) $T(n) \leq aT(n/b) + c(n^k log^p n)$

In (1) we change $log$ to $log_b$ by using the formula $log(n) = log(b)log_b(n)$. We get (2).

(2) $T(n) \leq aT(n/b) + clog^p(b)n^k(log_b n)^p$

Now we apply (2) to $n = b^m, b^{m-1}, \ldots, b^1$. We have the derivation below.

$T(b^m) \leq aT(b^{m-1}) + cb^{km}log^p(b)m^p$

$\leq a^2T(b^{m-2}) + cab^{k(m-1)}log^p(b)(m-1)^p + cb^{km}log^p(b)m^p$

$\leq a^2T(b^{m-3}) + ca^2b^{k(m-2)}log^p(b)(m-2)^p + cab^{k(m-1)}log^p(b)(m-1)^p + cb^{km}log^p(b)m^p$

$\leq \ldots$ // after m steps

$\leq a^mT(1) + clog^p(b)[a^{m-1}b^k1^p + a^{m-2}b^{2k}2^p + \ldots a^1b^{k(m-1)}(m-1)^p + a^0b^{km}m^p]$

So, we got (3), where $c_1 = clog^p(b)$.

(3) $T(b^m) \le a^m T(1) + c_1[a^{m-1}b^k 1^p + a^{m-2}b^{2k}2^p + \dots a^1 b^{k(m-1)}(m-1)^p + a^0 b^{km} m^p]$

We evaluate the sum (*).

(*) $S = a^{m-1}b^k 1^p + a^{m-2}b^{2k}2^p + \dots a^1 b^{k(m-1)}(m-1)^p + a^0 b^{km} m^p$

Wee have 3 cases.

**Case 1:** $a > b^k$.

In (*) we factor out $a^m$ and have (4).

(4) $S = (a^m/a)[(b^k/a)1^p + (b^k/a)^2 2^p + \dots (b^k/a)^{m-1}(m-1)^p + (b^k/a)^m m^p]$

All terms of the sum are positive. All we need to do is to show that the sum is bounded.

So, let's consider the series $U = (b^k/a)1^p + (b^k/a)^2 2^p + \dots (b^k/a)^{m-1}(m-1)^p + (b^k/a)^m m^p + \dots$.

The series is convergent by the ratio test because

$\lim_{m\to\infty} \frac{(b^k/a)^{m+1}(m+1)^p}{(b^k/a)^m m^p} = b^k/a < 1$

So, $U$ converges. Let $L$ be its sum. Then $S < L$, and

$T(b^m) \le a^m T(1) + (a^m/a)c_1 L = a^m(T(1) + c_1 L/a)$

Since $a^m = a^{\log_b(n)} = b^{\log_b a \log_b(n)} = n^{\log_b a}$

We have that $T(n) = O(n^{log_b a})$ holds for $n = b^m$.

**Case 2:** $a = b^k$.

Then the sum (4) becomes (5).

(5) $S = (a^m/a)[1^p + 2^p + \dots (m-1)^p + m^p]$

Let $R(m,p) = 1^p + 2^p + \dots (m-1)^p + m^p$.

Since $x^p$ is increasing on $[0,\infty)$ for $p > 0$,

$R(m,p) - m^p < \int_1^m x^p dx = (m^{p+1} - 1)/(p+1)$.

So, we have (6).

(6) $R(m,p) < m^p + m^{p+1}/(p+1) < m^{p+1} 2/(p+1)$

We replace (6) in (5) and have (7).

(7) $S < (a^m/a) 2m^{p+1}/(p+1) = 2a^m m^{p+1}/(a(p+1))$

We replace (7) in (3) and have the derivation below.

$T(b^m) < a^m T(1) + c_1 2a^m m^{p+1}/(a(p+1))$

$< a^m m^{p+1}(T(1) + 2c_1/(a(p+1)))$

$= b^{km} m^{p+1}(T(1) + 2c_1/(a(p+1)))$ // $a = b^k$

$= n^k log^{p+1}(n) log^{p+1}(b)(T(1) + 2c_1/(a(p+1)))$ // $n = b^m, m = lognlogb$

So, we have (8) where $c_2 = log^{p+1}(b)(T(1) + 2c_1/(a(p+1)))$.

(8) $T(n) < n^k log^{p+1}(n) c_2$

This tells us that $T(n) = O(n^k log^{p+1}(n))$.

**Case 3:** $a < b^k$

We evaluate (*) in a different way.

$S = a^{m-1}b^k 1^p + a^{m-2}b^{2k}2^p + \ldots a^1 b^{k(m-1)}(m-1)^p + a^0 b^{km} m^p$

$= b^{km}m^p((a/b^k)^{m-1}(1/m)^p + (a/b^k)^{m-2}(2/m)^p + \ldots + (a/b^k)^1((m-1)/m)^p + (a/b^k)^0(m/m)^p)$

$\leq b^{km}m^p((a/b^k)^{m-1} + (a/b^k)^{m-2} + \ldots + (a/b^k)^1 + (a/b^k)^0$ // since $p > 0$, $(i/m)^p \leq 1$ for all $1 \leq i \leq m$

$\leq b^{km}m^p \frac{1-(a/b^k)^m}{1-(a/b^k)}$ the sum of the geometric series

$< b^{km}m^p \frac{1}{1-(a/b^k)}$

We set $c_3 = \frac{1}{1-(a/b^k)}$ and get (9).

(9) $S \leq c_3 b^{km} m^p$

We use the inequality (9) in (3).

$T(n) = a^m T(1) + c_1 S < a^m T(1) + c_1 c_3 b^{km} m^p$

$\leq b^{km}m^p(T(1) + c_1 c_3)$ // $a^m < b^{km}m^p$

$= n^k log^p(n) log^p(b)(T(1) + c_1 c_3)$ // change $\log_b$ to $log$

So, we got (10), where $c_4 = (T(1) + c_1 c_3) log^p(b)$.

(10) $T(n) \leq c_4 n^k log^p(n)$.

**Part 4.2:** Assuming that the theorem holds for $n = b^m$ we will show that it holds for any $n$. For this we notice that $T(n)$ increases in $n$ and so are the 3 functions, $n^{log_b a}, n^k log^{p+1}(n), n^k log^p(n)$

The metodology is the same for all 3 cases. Every $n$ is sandwiched between two consecutive powers of $b$.

(11) $b^{m-1} \leq n < b^m$

So, we get (12).

(12) $n < b^m \leq bn$

Case 1: $T(n) \leq cn^{log_b a}$ for every power of $b$. Then we have the derivation below.

$T(n) \leq T(b^m)$ // $T$ is increasing

$\leq cb^{mlog_b a}$ // the inequqlity holds for $b^m$

$\leq c(b^m)^{log_b a}$

$\leq c(bn)^{log_b a}$ // the powers increase in the base and (11)

$\leq cb^{log_b a} n^{log_b a}$

$\leq can^{log_b a}$

This proves that $T(n)$ has the same Big Oh as $b^m$, but the constant is bigger.

Case 2: $T(n) \leq cn^k log^{p+1}(n)$ for all powers of $b$.

$T(n) \leq T(b^m)$ // $T$ is increasing

$\leq c(b^m)^k log^{p+1}(b^m)$ // the inequality holds for $b^m$

$\leq c(bn)^k log^{p+1}(bn)$ // the powers increase in the base and (12)

$\leq cb^k n^k 2^{p+1} log^{p+1}(n)$ // $n^2 \geq bn$ for $n \geq b$

$\leq cb^k 2^{p+1} n^k log^{p+1}(n)$

This proves that $T(n)$ has the same Big Oh as $b^m$, but the constant is bigger.

Case 3: $T(n) \leq cn^k log^p(n)$ for all powers of $b$.

The proof is the same as the one above.

**Grading Criteria:**

1. Part 4.1 : 20 points
2. Part 4.2: 15 points

**Question 5.** (20 points)

Solve the recurrence relation $T(n) = 7T(n-1) - 16T(n-2) + 12T(n-3)$ with the boundary conditions $T(0) = 2, T(1) = 7, T(2) = 19$.

**Solution:** The characteristic equation of the recurrence $T(n) = 7T(n-1) - 16T(n-2) + 12T(n-3) = 0$ is (1).

(1) $r^3 - 7r^2 + 16r - 12 = 0$

We can find its roots by using the cubic root formula or by trying to find a root among the divisors of 12. We are lucky and get the double root $r_1 = 2$ and $r_2 = 3$.

The solution is $T(n) = (An + B)2^n + C3^n$

We find the constants $A, B, C$ from the equations $T(0) = 2, T(1) = 7, T(2) = 19$.

$B + C = 2$

$2A + 2B + 3C = 7$

$8A + 4B + 9C = 19$

We get $A = 2, B = 3, C = -1$, so the solution is $T(n) = (2n + 3)2^n - 3^n$.

**Grading Criteria:**

1. If you got the roots you got 7 points

1. If you got $T(n) = (An + B)2^n + C3^n$ you have 14 points.

2. If you got the values of $A$, $B$, $C$ you get 6 more points.

**Question 6.** (25 points)

Solve the recurrence $T(n) = T(n-1) + n^4$ by finding a particular solution for the equation. Use $T(1) = 1$ as the boundary condition.

**Solution:** The solution of the equation (1) is $T(n) = T_h(n) + T_p(n)$ where $T_h(n)$ is the solution of the homogeneous equation (2) and $T_p(n)$ is a particular solution of (1).

(1) $T(n) = T(n-1) + n^4$

(2) $T(n) = T(n-1)$

$T_h(p) = A$, a constant.

For a particular solution we try $T_p(n) = Bn^5 + Cn^4 + Dn^3 + En^2 + Fn + G$.

We replace $T_p$ in (1) and get the identity (3).

(3) $Bn^5 + Cn^4 + Dn^3 + En^2 + Fn + G - B(n-1)^5 - C(n-1)^4 - D(n-1)^3 - E(n-1)^2 + F(n-1) - G \equiv n^4$

We expand the powers, move $n^4$ to the LHS, reduce the similar terms and get (4).

(4) $(5B-1)n^4 + (-10B+4C)n^3 + (10B-6C+3D)n^2 + (-5B+4C-3D+2E)n + (B-C+D-E+F) \equiv 0$

Since (4) holds for all $n$, the coefficients of the powers of $n$ must be zero. We get the system below.

$5B - 1 = 0$

$-10B + 4C = 0$

$10B - 6C + 3D = 0$

$-5B + 4C - 3D + 2E = 0$

$B - C + D - E + F = 0$

The solution to this system is $B = 1/5, C = 1/2, D = 1/3, E = 0, F = -1/30$.

We determine $A$ from the equation $T(1) = A + B + C + D + E + F$

We get $A = 0$. So,

$T(n) = n^5/5 + n^4/2 + n^3/3 - n/30 = n(n+1)(2n+1)(3n^2+3n-1)/30$

**Grading Criteria:**

1. If you got the system of equations in $B, C, D, E, F$ you got 15 points.
2. Solving the system is worth 8 points.
3. Finding $A$ is worth 2 points.
4. Just trying: 3 points

**Question 7.** (20 Points)

Prove by induction that that $T(n) = \lceil log(n+1) \rceil$ , where $T(n)$ is the largest number of comparisons needed to search an array with $n$ items sorted in increasing order using the binary search method. $\lceil x \rceil$, called the ceiling of $x$, is the smallest integer greater than or equal to $x$. For example, $\lceil 4.5 \rceil = 5$. Here $log$ is the logarithm in base 2.

**Solution 1:** First we notice that $\lceil log(n+1) \rceil = m+1$ iff $2^m \leq n < 2^{m+1}$.

We prove, by induction on $m$ that for all $n$ satisfying $2^m \leq n < 2^{m+1}$, $T(n) = m+1$.

Basis: $m = 0$.

Let $n$ satisfy the condition $2^0 \leq n < 2^1$. Then $1 \leq n < 2$, i.e. $n = 1$. Since $T(1) = 1$, we are done.

Inductive Step:

We assume that the statement is true for $m$ (the IH), and prove that it is true for $m+1$, i.e. we show (1).

(IH) For all $n$, $2^m \leq k < 2^{m+1}$ implies that $T(k) = m+1$.

(1) For all $n$, $2^{m+1} \leq n < 2^{m+2}$ implies that $T(n) = m+2$.

So let us assume (2).

(2) $2^{m+1} \leq n < 2^{m+2}$

The algorithm first checks the midpoint. Since we are looking for the worst case we assume that the inquiry is not found, and we have to search the largest of the 2 halves, $\lceil (n-1)/2 \rceil$. First we rewrite (2) as (3). Then we subtract -1 from all 3 terms and divide the differences by 2. We get (4).

(3) $2^{m+1} \leq n \leq 2^{m+2} - 1$

(4) $(2^{m+1} - 1)/2 \leq (n-1)/2 \leq (2^{m+2} - 2)/2$

We rewrite (4) as (5).

(5) $2^m - 1/2 \leq (n-1)/2 \leq 2^{m+1} - 1$

We take the ceiling of all terms in (5) and obtain (6).

(6) $\lceil 2^m - 1/2 \rceil \leq \lceil (n-1)/2 \rceil \leq \lceil 2^{m+1} - 1 \rceil$

We evaluate the ceiling and replace $\leq 2^{m+1} - 1$ by $< 2^{m+1}$ and get (7).

(7) $2^m \leq \lceil (n-1)/2 \rceil < 2^{m+1}$

The latest inequqlity allows us to apply the IH. So,

$T(n) = 1 + T(\lceil (n-1)/2 \rceil)$

$= 1 + (m+1) = m+2$.

**Solution 2:** Let us try a direct proof.

Basis: $n = 1$

Then $T(1) = 1$ and $\lceil log(1+1) \rceil = \lceil log2 \rceil = \lceil 1 \rceil = 1$.

Inductive Step: Assume that $T(m) = \lceil log(m+1) \rceil$ for all $1 \leq m \leq n$. Let us prove that it is true for $n+1$.

$T(n+1) = 1 + T(\lceil n/2 \rceil)$ the definition of $T(n)$
$= 1 + \lceil log(\lceil n/2 \rceil + 1) \rceil$ by IH
$= 1 + \lceil log(\lceil (n+2)/2 \rceil) \rceil$ push the second 1 under $\lceil\rceil$
$= \lceil 1 + log(\lceil (n+2)/2 \rceil) \rceil$ push the first 1 under $\lceil\rceil$
$= \lceil log2(\lceil (n+2)/2 \rceil) \rceil$ push 1 inside $log$

Now we consider the cases $n$ even and $n$ odd.

Case 1: $n = 2p$ Then we have the derivation
$\lceil log2(\lceil (n+2)/2 \rceil) \rceil = \lceil log2(\lceil (2p+2)/2 \rceil) \rceil$ $n = 2p$
$= \lceil log2(\lceil p+1 \rceil) \rceil$ because $(2p+2)/2 = p+1$
$= \lceil log2(p+1) \rceil$ because $\lceil p+1 \rceil = p+1$
$= \lceil log((2p+1)+1) \rceil$
$= \lceil log((n+1)+1) \rceil$ $n = 2p$

So, the formula holds for $n+1$.

Case 2: $n = 2p+1$
$\lceil log2(\lceil (n+2)/2 \rceil) \rceil = \lceil log2(\lceil (2p+3)/2 \rceil) \rceil$ because $n = 2p+1$
$= \lceil log2(p+2) \rceil$ since $\lceil (2p+3)/2 \rceil = p+2$
$= \lceil log(n+3) \rceil$ because $n = 2p+1$

But this is not what we wanted! We need $log(n+2)$. So, we have to show that for $n = 2p+1$, $\lceil log(n+3) \rceil = \lceil log(n+2) \rceil$.

So, we will show that $\lceil log(2p+4) \rceil = \lceil log(2p+3) \rceil$

Since both $log$ and $\lceil\rceil$ are increasing $\lceil log(2p+3) \rceil \leq \lceil log(2p+4) \rceil$. The inequality is strict only when there is some $m$ such that $2p+3 = 2^m$. But this imposible because $2p+3$ is odd and $@^m$ is even.

So, for $n = 2p+1$, $\lceil log(n+3) \rceil = \lceil log(n+2) \rceil$ and we can continue the proof.

$= \lceil log(n+3) \rceil = \lceil log(n+2) \rceil$
$= \lceil log((n+1)+1) \rceil$

**Grading Criteria:**

1. 3 points for trying