MIDTERM EXAM I SOLUTIONS
CSCI 163/COEN 179: THEORY OF ALGORITHMS
SPRING 2013

1. (10 points) Find an exact formula $M(n)$ for the number of multiplications performed by the following algorithm on input $n \geq 0$:

```
int q1(int n)
{
    int ans = 1;
    for (int i = 1; i <= n; ++i)
        for (int j = 0; j < n; ++j)
            for (int k = 0; k < i + j; ++k)
                ans += (i * j * k);
    return ans;
}
```

Answer:

$$
\begin{aligned}
M(n) &= \sum_{i=1}^{n}\sum_{j=0}^{n-1}\sum_{k=0}^{i+j-1}(2) \\
&= \sum_{i=1}^{n}\sum_{j=0}^{n-1}2((i+j-1)-0+1) \\
&= \sum_{i=1}^{n}\sum_{j=0}^{n-1}2(i+j) \\
&= 2\sum_{i=1}^{n}(\sum_{j=0}^{n-1}i+\sum_{j=0}^{n-1}j) \\
&= 2\sum_{i=1}^{n}(i\sum_{j=0}^{n-1}(1)+\frac{(n-1)n}{2}) \\
&= 2\sum_{i=1}^{n}(ni+\frac{(n-1)n}{2}) \\
&= 2(\sum_{i=1}^{n}(ni)+\sum_{i=1}^{n}(\frac{(n-1)n}{2})) \\
&= 2(\frac{n(n)(n+1)}{2}+\frac{n(n)(n-1)}{2}) \\
&= 2n^3
\end{aligned}
$$

2. (10 points) Find an exact closed-form formula $M(n)$ for the number of multiplications performed by the following algorithm on input $n \geq 0$:

```
int q2(int n)
{
    int ans = 1;

    if (n == 0)
        return ans;

    for (int j = 0; j < 10; j += 3)
        ans += n * j;
    return ans + q2(n-1);
}
```

Answer: The recurrence for $M(n)$ is:

$$\begin{aligned} M(0) &= 0 \\ M(n) &= 4 + M(n-1), \quad n \geq 1 \end{aligned}$$

Computing $M(n)$ for small values of n, we have

$$\begin{aligned} M(0) &= 0 \\ M(1) &= 4 + M(0) = 4 \\ M(2) &= 4 + M(1) = 8 \\ &\cdots \end{aligned}$$

We guess that $M(n) = 4n$ for $n \geq 0$. Verifying:

$$\begin{aligned} M(0) &= 4(0) \\ &= 0 \checkmark \\ 4 + M(n-1) &= 4 + 4(n-1) \\ &= 4n \\ &= M(n) \checkmark \end{aligned}$$

3. (10 points) Consider the following recurrence:

$$
\begin{aligned}
M(1) &= 1 \\
M(n) &= 4M(\frac{n}{4}) + \Theta(n^d), \quad n = 4, 16, 64, \ldots
\end{aligned}
$$

Use the master theorem to find the asymptotic growth rate of $M(n)$ when

(a) $d = 0$.
Answer: $4 = a > b^d = 4^0 = 1$, so $M(n) \in \Theta(n^{\log_4 4}) = \Theta(n)$

(b) $d = 1$.
Answer: $4 = a = b^d = 4^1$, so $M(n) \in \Theta(n^1 \lg n) = \Theta(n \lg n)$.

(c) $d = 2$.
Answer: $4 = a < b^d = 4^2$, so $M(n) \in \Theta(n^2)$.

4. (10 points) Which function grows faster ? Justify your answer.

$$n^3 \quad vs. \quad 8^{\lg n}.$$

Answer:

They grow at the same rate, since

$$\begin{aligned} 8^{\lg n} &= (2^3)^{\lg n} \\ &= (2^{\lg n})^3 \\ &= n^3. \end{aligned}$$

5. (10 points) Write an $o(n^2)$-time algorithm to solve the following problem:

   INPUT: an array $A[0..n-1]$ of real numbers, where $n \geq 2$.

   OUTPUT: the smallest difference between two different elements of A.

   Answer: A pair of elements of A whose values differ by the least amount must be adjacent when A is sorted. The following algorithm sorts A and finds the least difference between adjacent elements of A:

```
double q5(double A[], int n)
{
    merge_sort(A, 0, n-1);
    double ans = abs(A[1] - A[0]);

    for (int i = 2; i < n; ++i)
        ans = min(ans, abs(A[i]-A[i-1]));  // abs() and min() are
                                           //standard library functions

    return ans;
}
```

   The algorithm runs in time $\Theta(n \lg n) + \Theta(n) = \Theta(n \lg n)$ which is subquadratic as required.