**Problem 1.** ***DPV Problem 1.33.*** *How to compute:* By Wikipedia (so it must be right), $x \cdot y = \gcd(x,y) \cdot \operatorname{lcm}(x,y)$. So, $\operatorname{lcm}(x,y) = (x \cdot y)/\gcd(x,y)$.

*Time analysis:* If $x$ and $y$ are $n$-bits, then:

*(i)* the multiplication, $x \cdot y$, is $\Theta(n^2)$ time,
*(ii)* the gcd computation, $\gcd(x,y)$, is $O(n^3)$, and
*(iii)* the division $(x \cdot y)/\gcd(x,y)$ is $\Theta(n^2)$ time since $x \cdot y$ is at most $2n$ bits and $\gcd(x,y)$ is at most $n$ bits.

Hence, the total time is $O(n^3)$.

**Problem 2.** ***DPV Problem 1.36.***

**(a)** Since $p \equiv 3 \pmod 4$, we have $p+1 \equiv 4 \equiv 0 \pmod 4$. Hence, 4 divides $(p+1)$, and so $(p+1)/4$ is an integer.

**(b)** Suppose $p \equiv 3 \pmod 4$ and $x^2 \equiv a \pmod p$. Let $y = a^{(p+1)/4} \pmod p$. We want to show that $y^2 \equiv a \pmod p$. Note:

$$y^2 \equiv (a^{(p+1)/4})^2 \equiv a^{(p+1)/2} \equiv (x^2)^{(p+1)/2} \equiv x^{p+1} \pmod p.$$

But since $p+1 \equiv 2 \pmod{p-1}$, $x^{p+1} \equiv x^2 \equiv a \pmod p$. Therefore, $y$ is a square root of $a$ modulo $p$.

**Problem 3.** ***DPV Problem 1.45(b).*** We simply take

$$\text{verify}((N,e),s,M) = \begin{cases} \text{valid}, & \text{if } M = (s^e \bmod N); \\ \text{invalid}, & \text{otherwise.} \end{cases}$$

If $s = M^d \pmod N$, then by the correctness proof for RSA we know that $s^e \equiv (M^d)^e \equiv M^{d \cdot e} \equiv M \pmod N$. Computing $s^e \bmod N$ is $O(n^3)$ time. Hence, the entire implementation is $O(n^3)$ time.

**Problem 4.** ***DPV Problem 2.5.*** (**MT** = the Master Theorem)

**(a)** $\boxed{T(n) = 2T(n/3)+1.}$ So $a=2$, $b=3$, and $d=0$. Thus $0 = d < \log_b a = \log_3 2$, and by **MT**, $T(n) \in O(n^{\log_3 2})$.

**(b)** $\boxed{T(n) = 5T(n/4)+n.}$ So $a=5$, $b=4$, and $d=1$. Thus $1 = d < \log_b a = \log_4 5 \approx 1.161$, and by **MT**, $T(n) \in O(n^{\log_4 5})$.

**(c)** $\boxed{T(n) = 7T(n/7)+n.}$ So $a=7$, $b=7$, and $d=1$. Thus $1 = d = \log_7 7 = 1$, and by **MT**, $T(n) \in O(n \log n)$.

**(d)** $\boxed{T(n) = 9T(n/3)+n^2.}$ So $a=9$, $b=3$, and $d=2$. Thus $2 = d = \log_3 9 = 2$, and by **MT**, $T(n) \in O(n^2 \log n)$.

**(e)** $\boxed{T(n) = 8T(n/2)+n^3.}$ So $a=8$, $b=2$, and $d=3$. Thus $3 = d = \log_b a = 3$, and by **MT**, $T(n) \in O(n^3 \log n)$. Corrected

(i) $\boxed{T(n) = T(n-1)+c^n}$, where we take $T(0)=1$ and where $c>1$. First, expand out the recurrence to see if there is an obvious pattern. $T(n) = c^n + T(n-1) = c^n + c^{n-1} + T(n-2) = \cdots = c^n + c^{n-1} + \cdots + c + 1 = \frac{c^{n+1}-1}{c-1}$.‡ Let us verify by induction that: $T(n) = \frac{c^{n+1}-1}{c-1}$.

For $n=0$: $T(n) = 1 = \frac{c-1}{c-1} = \frac{c^{n+1}-1}{c-1}$.

For $n+1$: Suppose **IH**: $T(n) = \frac{c^{n+1}-1}{c-1}$. *Goal:* Show $T(n+1) = \frac{c^{n+2}-1}{c-1}$.

Note: $T(n+1) = c^{n+1} + T(n)$, which by the **IH**, is $= c^{n+1} + \frac{c^{n+1}-1}{c-1} = \frac{c^{n+2}-c^{n+1}+c^{n+1}-1}{c-1} = \frac{c^{n+2}-1}{c-1}$. **QED**

‡See Math Fact (k) in the Big-O writeup.

**Problem 5.** ***DPV Problem 2.12.*** Let $T(n)$ be the number of lines the program prints. When $n=1$, it does not print at all, hence $T(1)=0$. When $n>1$, it prints $1+2T(n/2)$ lines, hence $T(n) = 2T(n/2)+1$. So $a=2$, $b=2$, $d=0$, and hence, $0 = d < \log_b a = \log_2 2 = 1$. Hence, by **MT**, $T(n) = O(n^{\log_2 2}) = O(n)$.

**Problem 6.** ***PG Problem 337.***

The program is buggy. Consider $A[0]=1$, $A[1]=2$, $x=2$, $\ell=0$, and $r=1$. Then $m = \lfloor (0+1)/2 \rfloor = 0$ and $2 = x < A[m] = 1$. So the recursive call is search$(A,x,m,r)$. But since $m = \ell$, this is an infinite recursion.