C241 Homework Assignment 8

1.

Estimate the performance of the *Bubble Sort* program

```
begin
for i from 1 to N − 1 by 1 do
   begin
   for j from 1 to i − 1 by 1 do
     if A[j] ≤ A[j + 1]
     then skip;
     else
       begin
       t := A[j];
       A[j] := A[j + 1];
       A[j + 1] := t
       end
   end
end
```

Solution

---

1. The conditional statement `if` A`[`j`]` $\leq A$`[`$j+1$`]` `then` .... Performs two operations in its test and
   (a) no operations (or one if you like) to perform the `skip` statement in the `then` branch, or
   (b) three assignments and two additions on the `else` branch.

   So we must conservatively estimate that the conditional performs seven operations.

2. The inner loop initializes j, then runs the loop test $j = i-1$, the increment j `:=` $j+1$, and the conditional $i-1$ times. Putting these together, the inner loop runs
$$1 + \sum_{j=1}^{i-1} [7 + 2] = 1 + 9(i - 1)$$
   operations.

3. Similarly, the outer loop initializes i, then runs the test $i = N - 1$, the increment i `:=` $i + 1$, and the inner loop $N - 1$ times. So it performs
$$1 + \sum_{i=1}^{N-1} [2 + 1 + 9(i - 1)]$$
   operations.

Simplifying,

$$\begin{aligned}
1+\sum_{i=1}^{N-1}[2+1+9(i-1)] &= 1+\sum_{i=1}^{N-1}[3+9i-9] && \text{(arithmetic)} \\
&= 1+\sum_{i=1}^{N-1}[9i-6] && \text{(arithmetic)} \\
&= 1+9\cdot\left[\sum_{i=1}^{N-1} i\right]-\left[\sum_{i=1}^{N-1} 6\right] && \text{(splitting } \Sigma\text{)} \\
&= 1+9\cdot\left[\sum_{i=1}^{N-1} i\right]-6(N-1) && \left(\textstyle\sum_{\ell}^{u} C = C(u-\ell+1)\right) \\
&= 1+9\cdot\frac{(N-1)N}{2}-6(N-1) && \left(\textstyle\sum_{i=1}^{m} i = \frac{m(m+1)}{2}\right) \\
&= 9\cdot\frac{N^2-N}{2}-6N+7) && \text{(simplifying, combining like terms)} \\
&= \frac{18N^2-30N+14}{2} && \text{(adding fractions)} \\
&= 9N^2-15N+7 && \text{(reducing to lowest terms)}
\end{aligned}$$

2. Define two functions, f and g, for which $f \notin O(g)$ and $g \notin O(f)$.

Solution

We need only define two function which never dominate each other. For example, define

$$f(n) = \begin{cases} 0 & \text{if } n \text{ is even} \\ n & \text{if } n \text{ is odd} \end{cases} \qquad g(n) = \begin{cases} n & \text{if } n \text{ is odd} \\ 0 & \text{if } n \text{ is even} \end{cases}$$

There are no N and C such that, for all $n \geq N$, $g(n) \leq C \cdot f(n)$ or $f(n) \leq C \cdot g(n)$.

3. Prove: *If* $f_1 \in O(g_1)$ *and* $f_2 \in O(g_2)$ *then* $f_1(n) \times f_2(n) \in O\big(g_1(n) \times g_2(n)\big)$

Solution

Proof. Since $f_1(n) \in O(g_1(n))$ and $f_2(n) \in O(g_2(n))$, by Definition 6.1Order Notation and Order Arithmeticdefinition.6.1,

(a) there exist N_1 and C_1 such that for all $n \geq N_1$, $f(n) \leq C_1 \cdot g_1(n)$.

(b) there exist N_2 and C_2 such that for all $m \geq N_2$, $f(m) \leq C_2 \cdot g_2(m)$.

Let N be the larger of N_1 and N_2, and let C be the larger of C_1 and C_2. Now for any $n \geq N$,

1. $f_1(n) \leq C \cdot g_1(n)$ by (a) and because $n \geq N \geq N_1$ and $C \geq C_1$.
2. $f_2(n) \leq C \cdot g_2(n)$ by (b) and because $n \geq N \geq N_2$ and $C \geq C_2$.

Since these numbers are all positive, we can muliply the inequalities to get

$$f_1(n) \cdot f_2(n) \leq (C \cdot g_1(n)) \cdot (C \cdot g_2(n)) = C^2 \cdot (g_1(n) \cdot g_2(n))$$

By Definition 6.1Order Notation and Order Arithmeticdefinition.6.1, $f_1(n) \cdot f_2(n) \in O(g_1(n) \cdot g_2(n))$ with witnesses N and C^2.

4. Is $2^n \in O(n!)$ or is $n! \in O(2^n)$?

SOLUTION

Let's try a few values. The table below was generated by a Scheme program.

n	$n!$	2^n
0	1	1
1	1	2
2	2	4
3	6	8
4	24	16
5	120	32
6	720	64
7	5040	128
8	40320	256
9	362880	512
10	3628800	1024

Evidently $2^n \in O(n!)$ with witnesses $N = 4$ and $C = 1$. We can prove this by induction:

Proposition. For all $n \geq 4$, $n! \geq 2^n$.

PROOF The proof is by induction with $H[k] \equiv k \geq 4 \Rightarrow k! \geq 2^k$.

BASE CASE. The base case, for $k = 4$ is shown in the table above.

INDUCTION STEP. Assume $k! \geq 2^k$. Then

$$\begin{array}{rcll} (k+1)! & = & (k+1) \cdot k! & \text{(expanding } (k+1)!\text{)} \\ & \geq & 2^k \cdot (k+1) & \text{(I.H.)} \\ & \geq & 2^k \cdot 2 & (2 < 4 \leq k) \\ & = & 2^{k+1} & \end{array}$$

This completes the induction step.

□

5. Prove: *If* $f(n) \in O(h(n))$ *and* $g(n) \in O(h(n))$ *then* $f(n) + g(n) \in O(h(n))$.

SOLUTION

If $f(n) \in O(h(n))$ then there exist witnesses N_f and C_f such that

$$\text{For all } n \geq N_f,\ f(n) \leq C_f \cdot h(n)$$

And if $g(n) \in O(h(n))$ then there exist witnesses N_g and C_g such that

$$\text{For all } n \geq N_g,\ g(n) \leq C_g \cdot h(n)$$

Let $N = \max(N_f, N_g)$ and $C = \max(C_f, C_g)$. Then if $n \geq N$,

$$\begin{aligned}
f(n) + g(n) &\leq C_f \cdot h(n) + g(n) && (f \in O(g) \text{ with witnesses } N_f,\ C_f) \\
&\leq C_f \cdot h(n) + C_g \cdot h(n) && (g \in O(g) \text{ with witnesses } N_g,\ C_g) \\
&\leq C \cdot h(n) + C \cdot h(n) && (C = \max(C_f, C_g)) \\
&= 2C \cdot h(n)
\end{aligned}$$

Thus $f + g \in O(h)$ with witnesses N and $2C$.

6. Consider the program to the right, which computes over values in $\mathbb{N}$. Define the outer loop's invariant assertion I_1 to be:

$$I_1 \equiv z \cdot x^y = A^B$$

and the inner loop's invariant assertion I_2 to be:

$$I_2 \equiv z \cdot x^y = A^B \wedge y \neq 0$$

Since z is initialized to 1, I_1 is true when the program first reaches the outer loop; and since $I_2 \equiv I_1 \wedge y \neq 0$, I_2 holds whenever the inner loop is reached. Assuming that I_1 and I_2 are loop invariants, by the Theorem on Loop Invariants we will have $I_1 \wedge y = 0$ when the program terminates and hence

$$z \cdot x^y = z \cdot x^0 = z \cdot 1 = A^B$$

as desired. So it remains to be proved that I_1 and I_2 are invariants for their respective loops. Prove this.

```
{x = A ∧ y = B}
begin
z := 1;
while y ≠ 0 do {I1}
  begin
  while even?(y) do {I2}
    begin
    y := y/2
    x := x * x
    end;
  z := z * x;
  y := y − 1
  end
end
{z = A^B}
```

HINT. *Start with the inner loop.*

SOLUTION

Proposition 1. I_2 is an invariant for the inner loop, that is, if the body of the inner loop is executed with $I_2 \wedge even?(y)$ then I_2 will again be true afterward.

PROOF Suppose that $z \cdot x^y = A^B$, and $y \neq 0$ is an even number. The body of the inner loop computes new values x' and y' for the variables:

$$\begin{aligned} x' &= x \cdot x \\ y' &= \frac{y}{2} \\ z' &= z \end{aligned}$$

Since $y \neq 0$ and y is even, $y = 2k$ for some k and $\frac{y}{2} = k$ is still in $\mathbb{N}$. We want to show that $z' \cdot x'^{y'} = A^B$ and $y' \neq 0$. Thus,

$$\begin{aligned} z' \cdot x'^{y'} &= z \cdot (x^2)^{\frac{2k}{2}} && \text{(equations above)} \\ &= z \cdot (x^2)^k && \text{(algebraic simplifaction)} \\ &= z \cdot x^{2k} && \text{(multiplying exponents)} \\ &= z \cdot x^y && \text{(since } y = 2k \text{ for some } k \\ &= A^B && (I_2 \text{ is assumed)} \end{aligned}$$

as desired.

Proposition 2. I_1 is an invariant for the outer loop, that is, if the body of the loop is executed with $I_1 \wedge y \neq 0$, then I_1 will again be true afterward.

PROOF Suppose that $z \cdot x^y = A^B$ and $y \neq 0$. Then the inner loop exectutes and, by the Theorem on Loop Invariants, at the point where the inner loop terminates.

1. y is an odd number ($even?(y)$ is *false*), hence it also remains the case the $y \neq 0$.

2. I_2 is true, that is $x^y = A^B$ and $y \neq 0$

The remainder of the outer loop body computes values

$$\begin{aligned} z' &= z \cdot x \\ y' &= y - 1 \quad (\text{since } y \neq 0,\, y' \in \mathbb{N}) \\ x' &= x \ (x\text{'s value has not changed}) \end{aligned}$$

We need to show that I_1 is true, that is, $z' \cdot x'^{y'} = A^B$:

$$\begin{aligned} z' \cdot (x')^{(y')} &= z' \cdot x^{(y')} && (x' = x) \\ &= z' \cdot x^{(y-1)} && (y' = y - 1) \\ &= (z \cdot x) \cdot (x)^{(y-1)} && (z' = z \cdot x) \\ &= z \cdot \left[x \cdot (x)^{(y-1)}\right] && (\text{associativity}) \\ &= z \cdot x^y && (x \cdot x^{y-1} = x^y) \\ &= A^B && (\textit{assumption}) \end{aligned}$$

as needed.

Supplemental Problem. (Hotel Paradox)
[*This problem appears in many forms in many places.*]

Three grad students go to a conference and decide to save money by sharing a hotel room. On checking in, the desk clerk says the charge for a room containing two beds and a cot is \$30. After the students have paid, the desk clerk realizes that the charge should have been only \$25. She gives the doorkeeper a \$5 bill and asks that it be refunded to the students.

On the way to the room the doorkeeper realizes that he cannot divide \$5 by 3 evenly, so he decides to refund the students \$1 each and keeps \$2 for himself.

The students are happy that they paid only \$9 each for the room. The doorkeeper is happy that he gets a \$2 "tip." But $3 \times \$9 + \$2 = \$29$. Where did the other \$1 go?

Solution

The question is deceptively worded. There is no reason to add the net payment of \$27 to the doorkeeper's \$2 tip. Since the room costs \$25, the students have already, unkowingly, paid the tip. Their net cost is \$30 minus the \$5 refund *plus* the \$2 tip, totaling \$27.