Practice Quiz 1 Solutions

Problem PQ1-1. Integer Multiplication [**15 points**]

$$A^{(0)}B^{(0)} = A^{(k)}B^{(k)} + P^{(k)}.$$

Problem PQ1-2. True or False, and Justify

False. Observe that $\lg(n!) = \Theta(n \lg n) = O(n^{\lg_{99} 100 - 0.002})$. Therefore, by Master Theorem case 1, the solution is $\Theta(n^{\lg_{99} 100})$.

True. The correctness of radix sort requires the auxiliary sort to be stable. Insertion sort as presented in this course is stable.

False. All you know about the smallest item is that it must be on the bottom level (in the last half of the array). But it could be anywhere on the bottom level, so you'd have to look at the entire bottom level to find it. This takes $\Omega(n)$ time.

False. RANDOMIZED-SELECT was the first selection algorithm presented in Lecture 7. It is a randomized algorithm. The running time of a randomized algorithm is a random variable whose value for a particular run of the algorithm is determined by the random numbers the algorithm uses for that run. The running time of RANDOMIZED-SELECT does not does not depend its input (if all numbers are distinct). No adversary has control over which random numbers the algorithm will use, and no adversary can determine which random numbers the algorithm will use. Therefore, although it is true that RANDOMIZED-SELECT runs in $\Theta(n^2)$ time in the worst case, no adversary can force this behavior.

False. The procedure BUILD-HEAP runs in $\Theta(n)$ time. BUILD-HEAP was presented in Lecture 5, is given in CLR, and was part of Problem 3 of Problem Set 2.

An acceptable justification is that building a heap of n elements does not sort them, so the sorting lower bound does not apply.

True. As shown in Lecture 6, the number of leaves of a decision tree which sorts 6 elements is 6! and the height of the tree is at least lg(6!). Since $6! = 720$ and $2^9 = 512$ and $2^{10} = 1024$, we have $9 < \lg(6!) < 10$. Thus at least 10 comparisons are needed.

True.

1. Find the $\sqrt{n}th$ smallest element, r, using the SELECT algorithm.
2. Partition the array around r.
3. Sum the first $\sqrt{n}$ elements of the array.

Step 1 takes $\Theta(n)$ time, Step 2 takes $\Theta(n)$ time, and Step 3 takes $\Theta(\sqrt{n})$ time. Hence this gives a $\Theta(n)$ algorithm for summing up the smallest $\sqrt{n}$ elements in the given array.

True. A hash family $\mathcal{H}$ that maps a universe of keys U into m slots is *universal* if for each pair of distinct keys $x, y \in U$, the number of hash functions $h \in \mathcal{H}$ for which $h(x) = h(y)$ is exactly $|\mathcal{H}|/m$. In this problem, $|\mathcal{H}| = 3$ and $m = 3$. Therefore, for any pair of the four distinct keys, exactly 1 hash function should make them collide. By consulting the table above, we have:

$$\begin{array}{lll}
h(A) = h(B) & \text{only for } h_3 & \text{mapping into slot 2} \\
h(A) = h(C) & \text{only for } h_2 & \text{mapping into slot 0} \\
h(A) = h(D) & \text{only for } h_1 & \text{mapping into slot 1} \\
h(B) = h(C) & \text{only for } h_1 & \text{mapping into slot 0} \\
h(B) = h(D) & \text{only for } h_2 & \text{mapping into slot 1} \\
h(C) = h(D) & \text{only for } h_3 & \text{mapping into slot 0}
\end{array}$$

False. $\mathrm{E}[X(1-X)] = 0$, which can be seen in many ways:

- $X(1-X) = 0$ if $X = 1$ or $X = 0$.
- $\mathrm{E}[X(1-X)] = \mathrm{E}[X - X^2] = \mathrm{E}[X] - \mathrm{E}[X^2] = \mathrm{E}[X] - \mathrm{E}[X] = 0$.
- $\mathrm{E}[X(1-X)] = \sum_x x(1-x)\Pr\{X = x\} = 0(1-0)\Pr\{X = 0\} + 1(1-1)\Pr\{X = 1\} = 0$.

True. Consider writing the numbers in base n. They will have at most $\log_n m$ digits. We know that radix sort can sort n numbers with d digits where each digit is in the range $0..k$ in $O((n+k)d)$ time. So in this case, that is $O((n+n)\log_n m) = O(n \log_n m)$ time.

Problem PQ1-3. Pop Count

(a) The output of the stage-i adder must carry the addition of $i+1$ bits, which is a value of at most $i+1$ since all the bits in the sequence $\langle a_0, a_1, \cdots, a_i \rangle$ may be 1. As was stated in the problem, a cable of w wires can carry numbers up to $2^w - 1$. We therefore have the following equality and need to solve for w:

$$\begin{aligned}
i+1 &= 2^w - 1 \\
i+2 &= 2^w \\
\lg(i+2) &= w
\end{aligned}$$

Since $\lg(i+2)$ describes the number of wires needed to carry at most the value $i+1$, we must take the ceiling of it. Therefore,

$$w = \lceil \lg(i+2) \rceil = \Theta(\lg i)$$

wires are needed to carry the output of the stage-i adder.

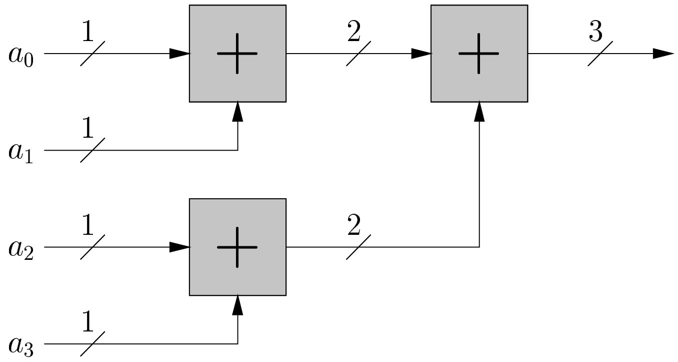
(b) Assume that, in order to add the first $n-1$ bits ($\langle a_0, a_1, \cdots, a_{n-2}\rangle$), the circuit uses $W(n-1)$ wires total in its $n-2$ stages. Then, in order to add in the last bit, we add one adder to the circuit, whose one input is the output of the last adder of the $n-1$ circuit. The other input to the new stage-$n-1$ adder is the last bit a_{n-1}. From part **(a)** we know that this new adder will require $\Theta(\lg(n-1)) = \Theta(\lg n)$ output wires. The recurrence that describes the total number of wires then becomes

$$\begin{aligned} W(n) &= W(n-1) + \Theta(\lg n) + 1 \\ &= W(n-1) + \Theta(\lg(n) + 1) \\ &= W(n-1) + \Theta(\lg n). \end{aligned}$$

The solution to this recurrence is obtained via *iteration*. Iteration gives us a Θ bound which in turn gives us an Ω bound.

$$\begin{aligned} W(n) &= W(n-1) + \Theta(\lg n) \\ &= W(n-1) + \Theta(\lg(n-1)) + \Theta(\lg(n)) \\ &= W(n-2) + \Theta(\lg(n-2)) + \Theta(\lg(n-1)) + \Theta(\lg(n)) \\ &\cdots \\ &= W(1) + \Theta(\lg(2)) + \Theta(\lg(n-2)) + \Theta(\lg(n-1)) + \Theta(\lg(n)) \\ &= 1 + \Theta(\lg(2) + .. + \lg(n-2) + \lg(n-1) + \lg(n)) \\ &= 1 + \Theta(\lg((n)(n-1)(n-2)..(2)(1))) \\ &= 1 + \Theta(\lg(n!)) \\ &= 1 + \Theta(n \lg n) \\ &= \Theta(n \lg n) \end{aligned}$$

We therefore conclude that $W(n) = \Omega(n \lg n)$.

(c) 

This circuit has $1+1+1+1+2+2+3 = 11$ **(eleven)** wires.

(d) The last stage of the divide-and-conquer pop-count circuit adds the output of two sized $n/2$ divide-and-conquer pop-count adders. That sum requires $\Theta(\lg n)$ wires to carry it (see part **(a)**). Therefore, the recurrence for the number of wires is

$$W(n) = 2W(n/2) + \Theta(\lg n).$$

The Master theorem case 1 gives the solution to this recurrence where

$$f(n) = \lg n = O(n^{\log_b a - \epsilon}) = O(n^{\log_2 2 - \epsilon}) = O(n)$$

for (say) $\epsilon = 1/2$. Thus,

$$W(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_2 2}) = \Theta(n).$$

Finally, we conclude our upper bound. $W(n) = \Theta(n) \Rightarrow W(n) = O(n)$.

(e) It's not clear where Blaise works, but it is clear that he doesn't know the material from 6.046. His iterative adder uses asymptotically more wires than the divide-and-conquer adder. In addition, his adder requires n stages to add while the divide-and-conquer adder has depth $\lg n$. Clearly, one of his graduate students saw the professor's initial design, suggested the improvement, and submitted the problem for use on the 6.046 quiz. [Answers along these lines were awarded one extra credit point.]