HW 4 Solutions

November 13, 2012

7-1

23.
We count the number of strings not containing 010 having i leading 0's. For $i = 0$, there are S_{n-1} such strings. For $i = 1$, the string begins 011, so there are S_{n-3} such strings. Similarly, for $i = 2$, there are S_{n-4} such strings; ... for $i = n - 3$, there are S_1 such strings. For $i = n - 2$, $n - 1$, or n, there is one such string. The equation now follows.

73.
1,5,2,4,3; 1,5,3,4,2; 2,5,1,4,3; 2,5,3,4,1; 3,5,1,4,2; 3,5,2,4,1; 4,5,1,3,2; 4,5,2,3,1; 1,3,2,5,4; 2,3,1,5,4; 1,4,2,5,3; 2,4,1,5,3; 1,4,3,5,2; 3,4,1,5,2; 2,4,3,5,1; 3,4,2,5,1; $E_5 = 16$

7-2

22. $a_n = 6\left(\frac{1}{3}\right)^n + 9n\left(\frac{1}{3}\right)^n$

23. Similar to Example 7.2.13, except for the initial conditions. To satisfy the initial conditions, we must have

$$\begin{aligned} b\left(\frac{1+\sqrt{5}}{2}\right) + d\left(\frac{1-\sqrt{5}}{2}\right) &= 1 \\ b\left(\frac{1+\sqrt{5}}{2}\right)^2 + d\left(\frac{1-\sqrt{5}}{2}\right)^2 &= 3 \end{aligned}$$

Solving the equation for d and b, we obtain, $b = d = 1$. Therefore, an explicit formula for the Lucas sequence is

$$L_n = \left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n.$$

35. Taking the logarithm to the base 2 of both sides of the equation, we obtain

$$\lg a_n = \frac{1}{2}(\lg a_{n-2} - \lg a_{n-1})$$

If we let $b_n = \lg a_n$, we obtain

$$b_n = \frac{-b_{n-1}}{2} + \frac{b_{n-2}}{2}$$

The quadratic equation

$$t^2 + \frac{t}{2} - \frac{1}{2} = 0$$

has two roots, $\frac{1}{2}$ and -1. Thus there exist constants p and q such that

$$b_n = p\left(\frac{1}{2}\right)^n + q(-1)^n.$$

Now,

$$b_0 = \lg a_0 = \lg 8 = 3$$

and a similar calculation shows that $b_1 = -\frac{3}{2}$. Therefore,

$$\begin{aligned} 3 &= p + q \\ -\frac{3}{2} &= \frac{p}{2} - q \end{aligned}$$

Solving for p and q, we obtain $p = 1$ and $q = 2$. Thus

$$b_n = \left(\frac{1}{2}\right)^n + 2(-1)^n$$

and

$$a_n = 2^{b_n} = 2^{[(1/2)^n + 2(-1)^n]}.$$

7-3

12. (a) We use induction on the size n of the sequence to prove that *binary_search2* is correct when the input is a sequence of size n.

The Basis is $n = 0$. In this case, the sequence is empty, $i > j$, and the algorithm correctly returns 0 to indicate that *key* is not found.
Now assume that if a sequence of length less than n is input to *binary_search2*, the algorithm returns the correct value. Suppose that a sequence of length $n > 0$ is input to *binary_search2*. Since $i \leq j$, the algorithm proceeds to the line where it computes k. If *key* is at index k, the algorithm correctly returns k. If *key* is not at index k, since the sequence is sorted, *key*, if present, is either in $s_i, \cdots, s_{k-1}$ or $s_{k+1}, \cdots, s_j$, but not both. The algorithm then executes

$$k1 = \textit{binary_search2}(s, i, k-1, key)$$

If *key* is present in $s_i, \cdots, s_{k-1}$, by the inductive assumption, *binary_search2* returns the index where it is located. If *key* is not present in $s_i, \cdots, s_{k-1}$, by the inductive assumption, *binary_search2* returns 0. The value returned is stored in $k1$. The algorithm then executes

$$k2 = \textit{binary_search2}(s, k+1, j, key_)$$

If *key* is present in $s_{k+1}, \cdots, s_j$, by the inductive assumption, *binary_search2* returns the index where it is located. If *key* is not present in $s_{k+1}, \cdots, s_j$, by the inductive assumption, *binary_search2* returns 0. The value returned is stored in $k2$. It follows that if *key* is not present in $s_i, \cdots, s_j$, $k1 + k2 = 0$. Since the algorithm returns $k1 + k2$, it follows that the algorithm is correct.

(b) We define the worst-case time required by the algorithm to be the number of times the algorithm is invoked in the worst case for a sequence containing n items. Let a_n denote the worst-case time.
Suppose $n = 0$, that is $i > j$. In this case, there is one invocation; so $a_0 = 1$.
Now suppose that $n > 1$. In the worst case, the item will not be found at the line

$$\text{if}(key == s_k)$$

so the algorithm will be invoked twice more:

$$\begin{aligned} k1 &= \textit{binary_search2}(s, i, k-1, key) \\ k2 &= \textit{binary_search2}(s, k+1, j, key) \end{aligned}$$

By definition, the first invocation will require a total of $a_{\lfloor (n-1)/2 \rfloor}$ invocations, and the second invocation will require a total of $a_{\lfloor n/2 \rfloor}$ invocations. Thus we obtain the recurrence relation

$$a_n = 1 + a_{\lfloor (n-1)/2 \rfloor} + a_{\lfloor n/2 \rfloor}.$$

We use strong induction to show that

$$a_n \leq 3n + 1$$

for all $n \geq 0$), thus proving that $a_n = O(n)$. The Base case, $n = 0$, is readily verified. Now assume that $n > 0$. By the inductive assumption,

$$a_{\lfloor (n-1)/2 \rfloor} \leq 3\lfloor (n-1)/2 \rfloor + 1$$

and

$$a_{\lfloor n/2 \rfloor} \leq 3\lfloor n/2 \rfloor + 1$$

Now

$$\begin{aligned} a_n &= 1 + a_{\lfloor (n-1)/2 \rfloor} + a_{\lfloor n/2 \rfloor} \\ &\leq 1 + 3\lfloor (n-1)/2 \rfloor + 1 + 3\lfloor n/2 \rfloor + 1 \\ &= 3 + 3(\lfloor (n-1)/2 \rfloor + \lfloor n/2 \rfloor) \\ &= 3 + 3(n-1) = 3n < 3n + 1 \end{aligned}$$

We conclude by using strong induction to show that

$$a_n \geq n$$

for all $n \geq 0$, thus proving that $a_n = \Omega(n)$ and, therefore, $a_n = \Theta(n)$. The Base Case, $n = 0$ is readily verified. Now assume that $n > 0$. By the inductive assumption,

$$a_{\lfloor (n-1)/2 \rfloor} \geq \lfloor (n-1)/2 \rfloor$$

and

$$a_{\lfloor n/2 \rfloor} \geq \lfloor n/2 \rfloor$$

Now

$$\begin{aligned} a_n &= 1 + a_{\lfloor (n-1)/2 \rfloor} + a_{\lfloor n/2 \rfloor} \\ &\geq 1 + \lfloor (n-1)/2 \rfloor + \lfloor n/2 \rfloor \\ &= 1 + (n-1) = n. \end{aligned}$$

8-1

18. The graph is bipartite. $V_1 = \{v_1, v_3, v_4, v_6, v_8, v_9, v_{10}\}$, $V_2 = \{v_2, v_5, v_7\}$.

19. The graph is bipartite. $V_1 = \{Gre, Buf, Sho, Dou, Mud\}$, $V_2 = \{She, Wor, Cas, Gil, Lan\}$.

47. 5

8-2

35. When n is odd.

36. Both m and n must be even.

72. 1, since $dist(v, w) = 1$ for every pair of distinct vertices in K_n.

8-3

17. (e, d, a, b, c, e)

32. Yes, $(a, b, c, j, i, m, k, d, e, f, l, g, h)$

8-4

3. 10; (a, b, c, d, z)

8-5

6. If $V_1 = \{a, b\}$ and $V_2 = \{c, d, e\}$ are the vertex sets and the ordering is a, b, c, d, e, the adjacency matrix is

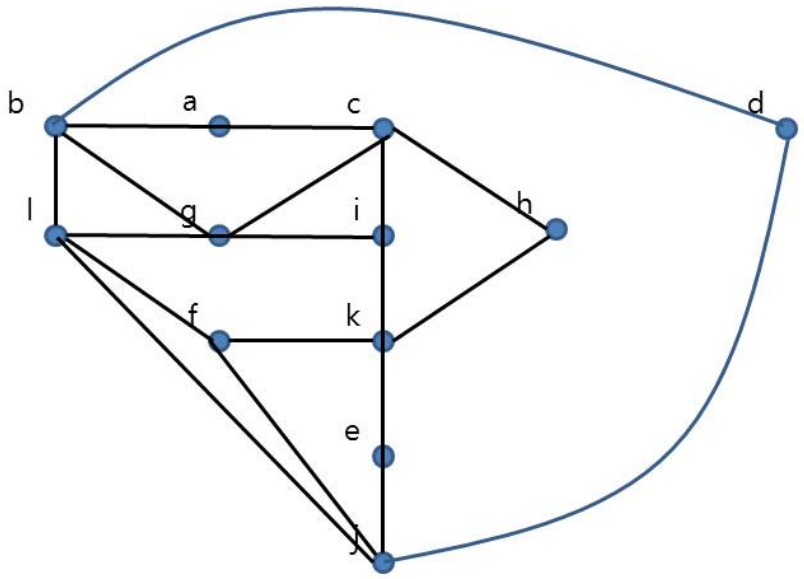
$$\begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

8-6

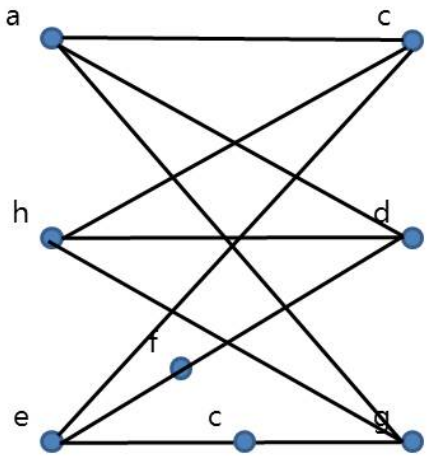
2. Relative to the vertex orderings a, b, c, d, e, f for G_1, and $4, 2, 1, 3, 6, 5$ for G_2, the adjacency matrices of G_1 and G_2 are equal.

8-7

3.



8. Not planar. The following graph is homeomorphic to $K_{3,3}$.



24. A-blue, B-green, C-red, D-yellow, E-green, F-red, G-yellow, H-green, I-yellow, J-green, K-blue, L-red.