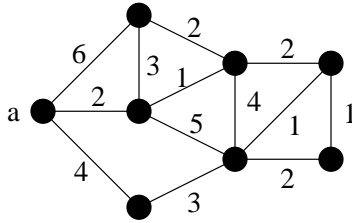
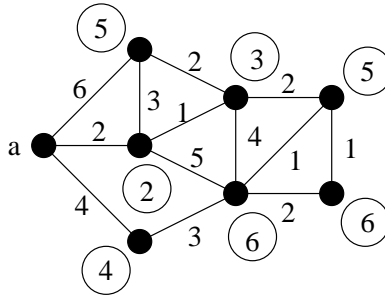


Final Exam Solutions

*Problem 1.* In the graph shown below, with edge lengths as indicated, find the length of the shortest path from vertex  $a$  to each of the other vertices.



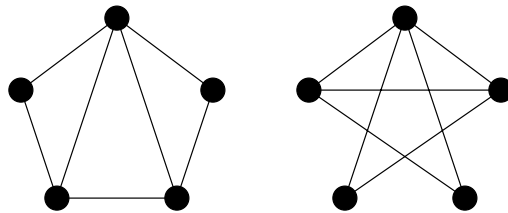
This can be solved using Dijkstra's algorithm or just by "inspection." The distances are circled on the diagram below.



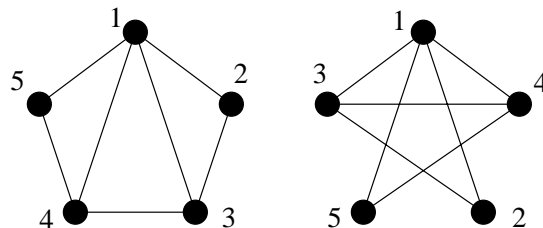
*Problem 2.* Prove that if ten points are picked in the interior of a square with sides of length 3, some two of the points must be no farther than  $\sqrt{2}$  apart.

Divide the big square into nine  $1 \times 1$  squares. By the pigeonhole principle, two of the points must be in the same  $1 \times 1$  square, so the farthest apart they could be is  $\sqrt{2}$ , the length of the diagonal of a unit square.

*Problem 3.* Either exhibit an isomorphism between the two graphs shown below, or prove that none exists.



The labels below exhibit an isomorphism.



*Problem 4.* Consider the function defined recursively for all positive integers  $n$  by  $f(1) = 1$  and  $f(n) = f(n - 1) + 2n - 1$  for  $n > 1$ . Find a simple formula for  $f(n)$  and prove that it is correct.

The formula is  $f(n) = n^2$ . We'll prove it by induction. For  $n = 1$  it is correct, since  $f(1) = 1$ . For  $n > 1$ , assume by induction that  $f(n - 1) = (n - 1)^2$ . Then using the definition of  $f(n)$ , we have

$$\begin{aligned} f(n) &= (n - 1)^2 + 2n - 1 \\ &= n^2 - 2n + 1 + 2n - 1 \\ &= n^2. \end{aligned}$$

*Problem 5.* Consider the following recursive algorithm for computing the  $n$ -th power of a square matrix  $A$ .

```

procedure power (square matrix  $A$ , positive integer  $n$ )
if  $n = 1$  return  $A$ 
else if  $n$  even
     $B := \text{power}(A, n/2)$ 
    return  $B \cdot B$ 
else
     $B := \text{power}(A, (n - 1)/2)$ 
    return  $A \cdot (B \cdot B)$ 

```

- (a) Prove that the algorithm is correct, *i.e.*, it always returns  $A^n$  for legal inputs  $A$  and  $n$ .  
 (b) Prove that the number of matrix multiplications required by the algorithm to compute  $A^n$  is less than or equal to  $2 \log_2(n)$ .

(a) Proof by induction. For  $n = 1$ ,  $\text{power}(A, 1)$  returns  $A$ , which is correct. For  $n > 1$  and even, we can assume by induction that  $\text{power}(A, n/2)$  returns  $A^{n/2}$ . Then  $B = A^{n/2}$  and  $\text{power}(A, n)$  returns  $B^2 = A^n$ . For  $n > 1$  and odd, we can assume that  $\text{power}(A, (n - 1)/2)$  returns  $A^{(n-1)/2}$ . Then  $B = A^{(n-1)/2}$  and  $\text{power}(A, n)$  returns  $AB^2 = A \cdot A^{n-1} = A^n$ .

(b) Method 1: Since  $n$  is cut at least in half on each recursive call, the algorithm invokes at most  $\log_2 n$  levels of recursion. At each level, one or two matrix multiplications occur, for a total of at most  $2 \log_2 n$ .

Method 2: Proof by induction. For  $n = 1$ ,  $\text{power}$  uses 0 matrix multiplications, which is equal to  $2 \log_2(1)$ . For  $n > 1$  and even, we can assume by induction that  $\text{power}(A, n/2)$  uses at most  $2 \log_2(n/2) = 2 \log_2(n) - 2$  multiplications. Then  $\text{power}(A, n)$  uses one more, for a total of  $2 \log_2(n) - 1$ , which is less than  $2 \log_2(n)$ . For  $n > 1$  and odd, we can assume that  $\text{power}(A, (n - 1)/2)$  uses at most

$$2 \log_2((n - 1)/2) = 2 \log_2(n - 1) - 2 < 2 \log_2(n) - 2$$

multiplications. Then  $\text{power}(A, n)$  uses two more for a total of less than  $2 \log_2 n$ .

*Problem 6.* How many ways are there to deal hands of 7 cards to each of 5 players, from a deck of 52 cards?

$$\binom{52}{7, 7, 7, 7, 7, 17}$$

*Problem 7.* Consider the single error correcting linear code over  $\mathbb{Z}_7$  with code matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 6 \end{bmatrix}.$$

(a) Which one of the following two vectors is a code vector, and how can you tell?

$$\mathbf{x} = [2 \ 1 \ 5 \ 3 \ 0 \ 2 \ 4 \ 1]$$

$$\mathbf{y} = [5 \ 3 \ 3 \ 0 \ 2 \ 2 \ 1 \ 0]$$

(b) Correct the error in the one that is not a code vector.

(a) Since the code matrix  $C$  has an identity matrix in the first six columns, the first six positions of a code vector are the message, and the last two positions can be computed by multiplying the message by the last two columns of  $C$ .

For the message  $[2 \ 1 \ 5 \ 3 \ 0 \ 2]$ , this gives  $[2 \ 1 \ 5 \ 3 \ 0 \ 2 \ 6 \ 1]$ , so  $\mathbf{x}$  is not a code vector.

For the message  $[5 \ 3 \ 3 \ 0 \ 2 \ 2]$ , this gives  $[5 \ 3 \ 3 \ 0 \ 2 \ 2 \ 1 \ 0]$ , so  $\mathbf{y}$  is a code vector.

(b) The code vector  $[2 \ 1 \ 5 \ 3 \ 0 \ 2 \ 6 \ 1]$  has Hamming distance 1 from  $\mathbf{x}$ , so it is the corrected vector.

*Problem 8.*

(a) Use Fermat's little theorem to compute  $5^{101} \pmod{11}$  and  $5^{101} \pmod{7}$ .

(b) Use part (a) and the Chinese remainder theorem to compute  $5^{101} \pmod{77}$ .

(a) Since 11 is prime,  $5^{10} \equiv 1 \pmod{11}$ , and  $5^{101} = 5 \cdot (5^{10})^{10} \equiv 5 \pmod{11}$ . Similarly,  $5^6 \equiv 1 \pmod{7}$  and  $5^{101} = 5^5 \cdot (5^6)^{16} \equiv 5^5 \equiv 3 \pmod{7}$ .

(b) With such small numbers, it is easiest just to try solutions of  $x \equiv 5 \pmod{11}$  until we find one that also solves  $x \equiv 3 \pmod{7}$ :  $x = 5$  no,  $x = 16$  no,  $x = 27$  no,  $x = 38$  yes. So  $5^{101} \equiv 38 \pmod{77}$ .

*Problem 9.* In a typical RSA cryptosystem, you encrypt a message  $x$  by computing  $y = x^e \pmod{n}$ , using the public key

$$n = 168199379620787065089088499614982334834522280642591021593331,$$

$$e = 52565485340775079244155772110124466324052187753568959049201.$$

(a) Explain briefly how you would decrypt the encrypted message  $y$  if you knew the private decryption key  $d$ .

(b) Explain briefly how you would find the private key  $d$  if you knew the prime factors of  $n$ .

(a) Compute  $x = y^d \pmod{n}$ .

(b) Let the prime factorization be  $n = pq$ . Then find the inverse  $d = e^{-1} \pmod{(p-1)(q-1)}$ , for instance using the Euclidean algorithm and back-substitution.

For the numbers given in the problem, it so happens that  $n$  is a product of two primes,  $e$  is relatively prime to  $(p-1)(q-1)$ , and the decryption key is

$$d = 38030557193149023861235413434311766695314554646776762250001.$$

*Problem 10.* Find a big- $\Theta$  estimate for the function

$$f(n) = 1^2 + 2^2 + 3^2 + \cdots + n^2,$$

as  $f(n) = \Theta(g(n))$ , where the function  $g(n)$  is given by a simple formula.

$$f(n) = \Theta(n^3).$$