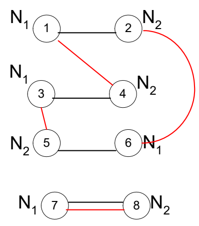
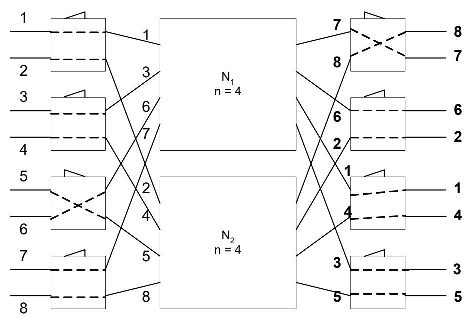
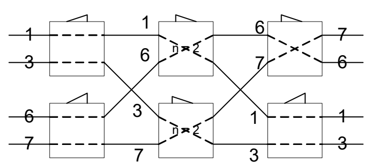
Solution #3:

1. Solution:

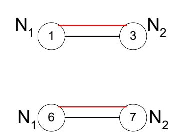


Forbidden Pairs Diagram

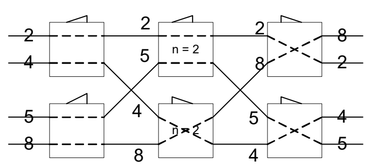




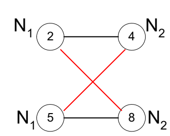
N_1



Forbidden Pairs Diagram



N_1

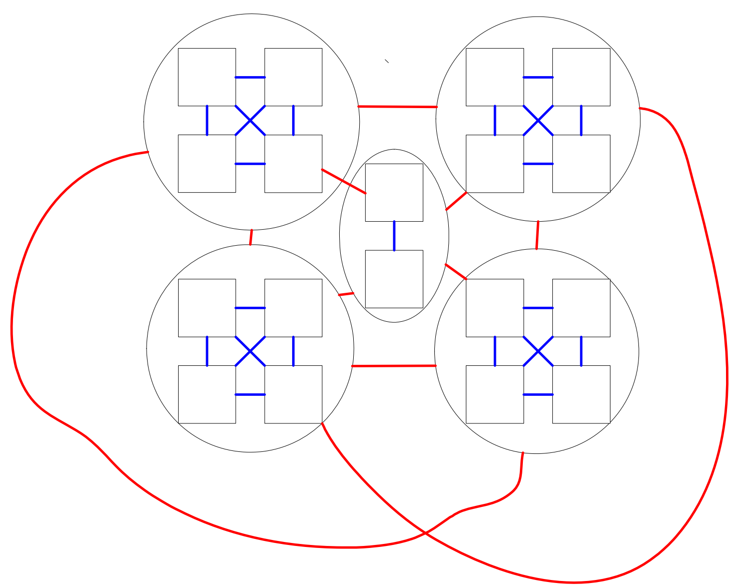


Forbidden Pairs Diagram

2. Problem 4.5

   Solution:

(a) It is possible to answer every pairwise test with bad-bad or good-good if the number of good chips is not more than half the total. [If the pair is "truly" good, then we have to answer good-good. If the pair is "truly" bad, then we may answer either good-good or bad-bad; indeed any answer is compatible but we give only these two. In case one of each, we can give bad-bad as the answer and be consistent.] So this is what an adversary will do. But from the adversary's answer we can not determine which chip is bad even though we may know that one is bad or which chip is good. So if we decide to throw a chip away, it might be the good one. If we throw one of these away, and keep one of them, whatever be the number of good chips, (as long it is not more than half the total), we will be forced to throw all of them away while attempting to find one good one. For example, suppose we know apriori that there are k good chips, even this information is not sufficient if the total number of chips is $2k$ or more as shown below. (the blue lines refer to "good-good" answers from the adversary and red lines indicate "bad-bad" answers. Let $n = rk + m; 0 \leq m < k; r \geq 2$. Partition the set $\{1, 2, ..., n\}$ into $r + 1$ sets the first r of which have k elements and the last has m elements. Let these sets be denoted by $S_i; i = 1, 2, ...(r + 1)$. Line joining (p, q) is blue if both p and q are in the same set of the partition and red otherwise. We can not find out which of the sets S_i contains good chips even though we know each set has either all good chips or all bad chips. Below is a graph for $r = 4; k = 4; m = 2$ The lines connecting groups represent several connections one for each pair of nodes in these sets.



So from now on we assume that the number of good chips is strictly greater than $\frac{n}{2}$ (this is the same as saying that the number of good chips is strictly greater than that of bad chips).

(b) Here is the complete solution: (I hope!)

Suppose we pair chips and test each pair: $\{1, 2\}, \{3, 4\}, ...(2\left\lfloor\frac{n}{2}\right\rfloor-1, 2\left\lfloor\frac{n}{2}\right\rfloor)\}$. Let T_1 be the set of pairs where exactly one of the pair says that the other is bad. If there is a pair $A - B$ in T_1 in which A says that B is good and B says that A is bad, then we know that A is truly bad (we can not conclude any thing about B). We can remove all such chips from further consideration and maintain the property that there are more good chips than bad ones. (But a clever adversary may produce answers that contain no such pairs). Let T_2 be the set of pairs where each chip says that the other is bad. We know that in each pair in T_2 there is at least one bad chip. We can **not** eliminate **one** of these and retain the other – because the chip that eliminated might be a good one and this might reduce the count of good ones. So we **must eliminate both such chips** out of these pairs in T_2. This still preserves the property that there are more good than bad chips in the set that remains. When we eliminate the set T_2 from further consideration, the remaining number may not be a power of 2 even if we started with a power of 2. So the assumption that the number of chips is a power of 2 is useless.

Let S be the set of pairs in the category where each chip says the other is good. [Each pair is in $T_1 \cup T_2 \cup S$ and these sets are not overlapping.] For each pair in S, either both are good or both are bad. So we can not eliminate both; we can not keep both. Eliminating both might reduce the number of good chips. keeping both might not reduce the set under consideration down to half the size of the original set. So we must eliminate one from each pair without knowing whether the pair is truly good or truly bad. But we have to be careful. In case n is odd, last chip is not tested in this round. We can either keep it for further consideration or not. But whatever we do, we need to make sure that in the set that remains there are more good than bad chips. For this purpose, we keep the last chip when n is odd if the number of pairs in S is even and do not keep it if this number is odd. If the number of pairs in S is even, then we could have the possibility of an equal number of truly good pairs and bad pairs. If we retain one from each pair, then the number of good and bad chips might be equal if we do not include the last chip when n is odd. Fortunately in this case, the last chip is truly good and so retaining it for further tests preserves the property that there are more good than bad. If the number of pairs is odd, the fact that there are more good than bad before elimination, tells us that the number of truly good pairs exceeds the number of truly bad pairs. So in this case we do not retain the last chip when n is odd. This is the basis of our divide and conquer strategy. This yields a recurrence relation:

$$t(n) \leq t(\left\lceil\frac{n}{2}\right\rceil) + \frac{n}{2}$$

Since we have eliminated at least roughly half the chips in this process by doing $\lfloor \frac{n}{2} \rfloor$ tests. This recurrence has a solution $t(n) = O(n)$.

Once a good chip is found (when we reach $n = 1$ or 2 as a boundary condition), then we can use this to check all others in n tests.

3. Problem 4-6:

   Solution:

   (a) An $m \times n$ array A is a Monge array if it satisfies the relations:

   $$A[i,j] + A[k,l] \leq A[i,l] + A[k,j] \qquad \forall i,j,k,l \text{ such that } i < k; j < l$$

   By taking $k = i + 1$ and $l = j + 1$ it follows that the above implies the relations

   $$A[i,j] + A[i+1,j+1] \leq A[i,j+1] + A[i+1,j] \qquad \forall i,j$$

   We also get if we let $k = i+1$ and $l = j+2$(with j replaced by $j+1$)

   $$A[i,j+1] + A[i+1,j21] \leq A[i,j+2] + A[i+1,j+1] \qquad \forall i,j$$

   Adding these two inequalities and canceling common terms we get

   $$A[i,j] + A[i+1,j+2] \leq A[i,j+2] + A[i+1,j+1] \qquad \forall i,j$$

   Use induction on the value of $k - i$ and $l - j$ (one inside the other) to show the above result.

   (b) Change $A[2,3]$ from 7 to 5

   (c) Suppose, by way of contradiction, that the statement

   $$f(1) \leq f(2) \leq ... \leq f(m)$$

   is false. Let $f(i+1) < f(i)$ for some i. Consider the inequality

   $$A[i,j] + A[k,l] \leq A[i,l] + A[k,j] \qquad \forall i,j,k,l \text{ such that } i < k; j < l$$

   with $j = f(i+1); k = i+1;$ and $l = f(i)$. We get

   $$A[i,f(i+1)] + A[i+1,f(i)] \leq A[i,f(i)] + A[i+1,f(i+1)]$$

   But $A[i,f(i)] < A[i,f(i+1)]; A[i+1,f(i+1)] \leq A[i+1,f(i)]$ since $f(p)$ is the left most minimum element in the p^{th} row of A. This leads to a contradiction and hence the result.

   (d) Suppose we know the values of $f(2i); 1 \leq i \leq \lfloor \frac{n}{2} \rfloor$. We can compute $f(2i-1)$ by checking $A[2i-1,j]; f(2i-2) \leq j \leq f(2i)$. The number of elements that we have to consider to compute all of these is no more than $m + n$. Hence the result.

(e)

$$
\begin{aligned}
t(m,n) &= t(\left\lceil \frac{m}{2} \right\rceil, n) + O(m+n) \\
t(1,n) &= n-1
\end{aligned}
$$

Use substitution method to verify that the solution is $t(n) = O(m + n \lg m)$

4. 9-2: Given n elements $x_1, x_2, ..., x_n$ with positive weights $w_1, w_2, ..., w_n$ such that $\sum_{i=1}^{n} w_i = 1$, the weighted (lower) median is the element x_k that satisfies the relations:

$$
\begin{aligned}
\sum_{i:x_i<x_k} w_i &< \frac{1}{2} \\
\sum_{i:x_i>x_k} w_i &\leq \frac{1}{2}
\end{aligned}
$$

   - We want to compute the weighted median in $\Theta(n)$ worst-case time.

     Solution:

     Find the regular median of the x_i. Let the index of of this element be i_1. Check the above two conditions with $k = i_1$. (Use PARTITION for this with this element as the pivot element). If they are satisfied, we have the required weighted median. If not either $\sum_{i:x_i<x_k} w_i \geq \frac{1}{2}$ or $\sum_{i:x_i>x_k} w_i > \frac{1}{2}$ (both can not happen – why?). In the first case, the weighted median is in the LOW side of the above PARTITION. In the second case, it is on the HIGH side. All elements on the other side can be condensed into one element with weight equal to the total weight of that side in the future steps. Since finding the median, PARTITION and condensing take $\Theta(n)$ worst-case time, the recurrence relation for this becomes:

$$
t(n) = t(\left\lceil \frac{n}{2} \right\rceil) + \Theta(n)
$$

     whose solution is $t(n) = \Theta(n)$.

5. 9.3-1:

   In SELECT algorithm, we grouped elements into sets of 5 each (except possibly the last one which may contain fewer). Analyze the algorithm if we made the following changes (one at a time)

   (a) We grouped them into sets of 7

   (b) We group them into sets of 3

   Justify your answer.

   Solution:

(a)) $|LOW| \geq \frac{4n}{14} - 8$; $|HIGH| \geq \frac{4n}{14} - 8$; hence $|LOW| \leq \frac{10n}{14} + 8$; $|HIGH| \geq \frac{10n}{14} + 8$. Therefore,

$$t(n) = t(\left\lceil \frac{n}{7} \right\rceil) + t(\frac{5n}{7} + 8) + cn$$

By substitution we can show that in this case $t(n) = O(n)$. So this actually does work similar to taking groups of 5.

(b) $|LOW| \geq \frac{n}{3} - 4$; $|HIGH| \geq \frac{n}{3} - 4$; hence $|LOW| \leq \frac{2n}{3} + 4$; $|HIGH| \geq \frac{2n}{3} + 4$. Therefore,

$$\begin{aligned} t(n) &= t(\left\lceil \frac{n}{3} \right\rceil) + t(\frac{2n}{3} + 4) + cn \\ &\geq t(\frac{n}{3}) + t(\frac{2n}{3}) + cn \end{aligned}$$

This is a particular case of the equation

$$t(n) = t(\alpha n) + t((1-\alpha)n) + cn; \qquad 0 < \alpha < 1$$

whose solution by recursion tree (iteration method) is $t(n) = \Theta(n \lg n)$ Hence in this case, we will not get a linear time algorithm.

6. Problem 9.3-8: Let $X[1, 2, ..., n]$ and $Y[1, 2, ..., n]$ be two sorted arrays, each containing n numbers. We want to find the median of the $2n$ numbers in arrays X and Y. How would you use divide and conquer method to solve this problem? Write down the recurrence relation that arises and using the master theorem, find the solution to this relation. {Hint: You are shooting for an $O(\lg n)$ algorithm overall}

   Solution:

   if $X[\lceil \frac{n}{2} \rceil] = Y[\lceil \frac{n}{2} \rceil]$ **then** the median is $X[\lceil \frac{n}{2} \rceil]$

   else if $X[\lceil \frac{n}{2} \rceil] < Y[\lceil \frac{n}{2} \rceil]$ find the median of the sorted arrays $X[\lfloor \frac{n}{2} \rfloor + 1, 2, ..., n]$ and $Y[1, 2, ..., \lceil \frac{n}{2} \rceil]$

   else find the median of the sorted arrays $X[1, 2, ..., \lceil \frac{n}{2} \rceil]$ and $Y[\lfloor \frac{n}{2} \rfloor + 1, 2, ..., n]$.

   Suppose we have $X[\lceil \frac{n}{2} \rceil] = Y[\lceil \frac{n}{2} \rceil]$. Then $X[i] \leq X[\lceil \frac{n}{2} \rceil] = Y[\lceil \frac{n}{2} \rceil] \leq X[j]$ for $i < \lceil \frac{n}{2} \rceil < j$ and $Y[i] \leq X[\lceil \frac{n}{2} \rceil] = Y[\lceil \frac{n}{2} \rceil] \leq Y[j]$ for $i < \lceil \frac{n}{2} \rceil < j$. Hence $X[\lceil \frac{n}{2} \rceil] = Y[\lceil \frac{n}{2} \rceil]$ is the overall median.

   Now suppose $X[\lceil \frac{n}{2} \rceil] < Y[\lceil \frac{n}{2} \rceil]$. Each element of the type $X[j]$ for $j < \lceil \frac{n}{2} \rceil$ is less or equal to at least n elements in the union and each element of the type $Y[j]$ for $j > \lceil \frac{n}{2} \rceil$ is greater or equal to at least n elements in the union. Hence none of these can be the choice of the overall median. So the algorithm is correct in eliminating them from further consideration. Similar arguments hold for the remaining case.

Since the problem size is reduced to half the size of the original problem, we have the recurrence relation:

$$t(n) = t(\left\lceil \frac{n}{2} \right\rceil) + 1$$

whose solution is $t(n) = \Theta(\lg n)$.