Problem 1. Book Search. Find an example of a "divide and conquer" algorithm in our textbook, outside Chapters 4 and 30. It could be in the main text, or in an exercise or problem. Briefly describe the problem, the algorithm, its recurrence, and time bound. Also give Chapter and page numbers. (Bonus: find one that nobody else does!)

In Chapter 9.3, page 219, there is a SELECT algorithm. This algorithm finds the desired element by recursively partitioning the input array by using the median of medians selection algorithm. I think this median of medians algorithm is a divide and conquer strategy. It has three steps.

1. It divides the n elements of an unsorted array into $\lfloor n / 5 \rfloor$ groups of 5 elements each and at most one group made up of the remaining n mod k elements.
2. Then we find and pick up the median of each group by first insert-sorting them.
3. Recursively find the median of the $\lfloor n / 5 \rfloor$ medians found in step 2.

Step 1 takes O(n) time to building new group. Step 2 takes $O(n)$ because insertion sort on sets of size $O(1)$. Step 3 takes time T($\lfloor n / 5 \rfloor$). So the time bound of this algorithm must be $O(n)$.

Problem 2

4-1 Recurrence Example

a. $T(n) = 2T(n/2) + n^4 \Rightarrow \because f(n) = n^4 = \Omega(n^{\log_2 2+\varepsilon})$ for $\varepsilon > 0$, such as $\varepsilon = 1, 2$,

and $2f(n/2) \le \frac{1}{2} f(n)$

so. $T(n) = \Theta(f(n)) = \Theta(n^4)$ - case 3

b. $T(n) = T(7n/10) + n. \Rightarrow \because f(n) = n = \Omega(n^{\log_{10/7} 1+\varepsilon})$ for $\varepsilon = 1$,

and $f(7n/10) \le \frac{7}{10} f(n)$

$\therefore$ $T(n) = \Theta(f(n) = \Theta(n)$ - case 3

c. $T(n) = 16T(n/4) + n^2 \Rightarrow \because f(n) = n^2 = \Theta(n^{\log_4 4^2})$

$\therefore$ $T(n) = \Theta(n^2 \lg^n)$ - case 2

d. $T(n) = 7T(n/3) + n^2 \Rightarrow \because f(n) = n^2 = \Omega(n^{\log_3 7+\varepsilon})$ for $\varepsilon = 1$.

and $7f(n/3) \le \frac{8}{9} f(n)$

$\therefore$ $T(n) = \Theta(f(n)) = \Theta(n^2)$ - case 3

e. $T(n) = 7T(n/2) + n^2 \Rightarrow \because f(n) = n^2 = O(n^{\log_2 7-\varepsilon})$ for $\varepsilon = 1$

$\therefore$ $T(n) = \Theta(n^{\log_2 7})$ - case 1

f. $T(n) = 2T(n/4) + \sqrt{n} \Rightarrow \because f(n) = n^{\frac{1}{2}} = \Theta(n^{\log_4 2})$

$\therefore$ $T(n) = \Theta(n^{\frac{1}{2}} \cdot \lg^n)$ - case 2

g. $T(n) = \cancel{T(n-2)} + n^2$. Master Theorem does not apply to this recurrence

$\cancel{T(n-2)} = \cancel{T(n-4)} + (n-2)^2$.

$T(\cancel{n-2k+2}) = T(n-2k) + (n-2k+2)^2$.

sum up these equation. $\Rightarrow$. $T(n) = \sum_{k=0}^{n/2} (n-2k)^2 = \Theta(n^3)$.

because the max order item must be $\frac{n}{2} \cdot n^2$.

Problem 3

4-3 More recurrence examples

a. $T(n) = 4T(n/3) + n\lg^n \Rightarrow \because f(n) = n\lg^n = O(n^{\log_3 4-\varepsilon})$ where $\varepsilon = 0.1$.
because $\lg^n = O(\log_3 \frac{4}{3})$
$\therefore T(n) = \Theta(n^{\log_3 4})$ — case 1

b. $T(n) = 3T(n/3) + n/\lg^n \Rightarrow \because f(n) = n/\lg^n$, and $n^{\log_b a} = n^{\log_3 3} = n$
$n^{\log_b a}/f(n) = n/(n/\lg^n) = \lg^n$,
$\lg^n$ is asymptotically less than n^ε for any positive constant ε.
$\therefore$ the theorem does not apply.
Then, $T(n) = 3T(n/3) + n/\lg^n = 3^2 T(n/3^2) + n/\lg^n + n/(\lg^{n/3})$
$= \cdots = 3^k T(n/3^k) + n \cdot \sum_{i=0}^{k-1} \frac{1}{\lg^{(n/3^i)}}$
$= 3^k T(n/3^k) + n \cdot \sum_{i=0}^{k-1} (\lg^n - i)^{-1}$ (do not consider $\underbrace{\text{coefficients}}_{\text{constant}}$) like $\lg^3$.
let $T(n/3^k) = T(1) = 1 . \Rightarrow n = 3^k$
$\therefore T(n) = n + n \cdot \sum_{i=0}^{k-1} (k-i)^{-1}$
According to Harmonic series, $T(n) \approx n(1 + \lg^k) = \Theta(n \cdot \lg^{(\lg^n)})$

c. $T(n) = 4T(n/2) + n^2\sqrt{n} \Rightarrow f(n) = n^2\sqrt{n}$, $n^{\log_2 4} = n^2$.
$\because f(n) = n^2\sqrt{n} = \Omega(n^{2+\varepsilon})$ where $\varepsilon = 0.1$
and $4f(n/2) \leq \frac{\sqrt{2}}{2} f(n)$. — case 3
$\therefore T(n) = \Theta(f(n)) = \Theta(n^2 \cdot \sqrt{n})$

d. $T(n) = 3T(n/3 - 2) + n/2 . \Rightarrow \because$ Compare to $n/3$, -2 maybe can be ignored.
$\therefore$ See $T(n)$ as $T(n) = 3T(n/3) + n/2$
In this case, $n^{\log_b a} = n$. $f(n) = \Theta(n)$ — case 2.
$\therefore T(n) = \Theta(n \cdot \lg^n)$.

e. $T(n) = 2T(n/2) + n/\lg^n \Rightarrow$ same as problem (b), the theorem does not apply.
Recursive method: $T(n) = 2^k T(n/2^k) + n \cdot \sum_{i=0}^{k-1} (\lg^n - i)^{-1}$,
let $T(n/2^k) = T(1) = 1$, $T(n) = n + n \sum_{i=0}^{k-1} (k-i)^{-1}$
$\Rightarrow T(n) = \Theta(n \cdot \lg^{(\lg^n)})$.

f. $T(n) = T(n/2) + T(n/4) + T(n/8) + n$.

The theorem does not apply. Use recursion Tree.

$$T(n) \quad \cdots\cdots \; n.$$

$$T(n/2) \quad T(n/4) \quad T(n/8) \quad \cdots\cdots \; \tfrac{7}{8}n.$$

$$T(n/4) \; T(n/8) \; T(n/16) \; T(n/8) \; T(n/16) \; T(n/32) \; T(n/16) \; T(n/32) \; T(n/64) \quad \cdots \; (\tfrac{7}{8})^2 n$$

The shortest branch $= n \to \frac{n}{8} \to \frac{n}{64} \to \cdots \to 1$

The longtest branch $= n \to \frac{n}{2} \to \frac{n}{4} \to \cdots \to 1$

Up to $\log_8^n$ levels, the tree is full, and then go down, some nodes are absent.

Upbound: $T(n) = O\left(\sum_{i=0}^{\log_2^{n}-1} (\frac{7}{8})^i \cdot n\right) = O(n)$

Low bound: $T(n) = \Omega\left(\sum_{i=0}^{\log_8^{n}} (\frac{7}{8})^i n\right) = \Omega(n)$ $\Rightarrow T(n) = \Theta(n)$

$\therefore$

g. $T(n) = T(n-1) + \frac{1}{n}$. $\Rightarrow$ The theorem does not apply.

$T(n) = T(n-1) + \frac{1}{n} = T(n-2) + \frac{1}{n} + \frac{1}{n-1} = T(n-k) + \sum_{i=0}^{k} \frac{1}{n-i}$

let $T(n-k) = T(1) = 1$, Then $T(n) = 1 + \sum_{i=0}^{n-1} \frac{1}{n-i}$.

According to Harmonic Series, $T(n) = \Theta(\lg^n)$

h. $T(n) = T(n-1) + \lg^n$. $\Rightarrow$ The theorem does not apply.

$T(n) = T(n-1) + \lg^n = T(n-2) + \lg^n + \lg^{n-1} = T(n-k) + \lg\left(\prod_{i=0}^{k}(n-i)\right)$

let $T(n-k) = T(1) = 1$. Then $T(n) = 1 + \lg\left(\prod_{i=0}^{n-1}(n-i)\right) = \Theta(n \lg^n)$

i. $T(n) = T(n-2) + \frac{1}{\lg^n}$ $\Rightarrow$ The theorem does not apply.

$T(n) = T(n-2) + \frac{1}{\lg^n} = T(n-4) + \frac{1}{\lg^n} + \frac{1}{\lg^{(n-2)}} = T(n-2k) + \sum_{i=1}^{k} \frac{1}{\lg^{2i}}$

let $T(n-2k) = T(1)$. Then $T(n) = \sum_{i=1}^{n/2} \frac{1}{\lg^{2i}} = \Theta(\lg(\lg^n))$

j. $T(n) = \sqrt{n} \cdot T(\sqrt{n}) + n.$ $\Rightarrow$ The theorem does not apply.

(P.S. I was struggling in this problem, so I looked into some solution online. Now I figured it out, and know how to solve this type of problems.)

let $n = C^{2^k}$, then $T(C^{2^k}) = C^{2^{k-1}} \cdot T(C^{2^{k-1}}) + C^{2^k}$

$$f_c(k) = C^{2^{k-1}} \cdot f_c(k-1) + C^{2^k} = C^{2^{k-1}} \cdot (C^{2^{k-2}} \cdot f_c(k-2) + C^{2^{k-1}}) + C^{2^k}$$

$$= l \cdot C^{2^k} + C^{(2^l - 1) \cdot 2^{k-l}} \cdot f_c(k-l)$$

l can go up to k to make $T(C^{2^{k-l}}) = T(1)$ $\therefore$

$$f_c(k) = k \cdot C^{2^k} + C^{2^k - 1} \cdot f_c(0) = (k+1) C^{2^k}$$

$\because$ $n = C^{2^k}$ $\Rightarrow$ $\log_c n = 2^k$ $\Rightarrow$ $k = \log_2(\log_c n)$

$\therefore$ $T(n) = \theta(n \cdot \lg(\lg n))$

Problem 4. Book Problem 4-5, Chip pages 109–110, three parts. This is the chip testing problem, give a brief answer for each part.

1. A good chip would say good if the other is good and say bad if the other is bad. A bad chip, would say good if the other is bad, and say bad if the other is good. So it is hard to discriminate them if the pair are both good or both bad because the act between bad chips is similar to the act between good chips. The only way we can do is to eliminate one bad chip exactly when the outcome is “at least one is bad.” Then we can cut down the size of bad chips until no bad chips left. But if the amount of good chips is less than n/2, the bad chip pairs must be more than good chip pairs. The good chips would not be left before no bad chips is left.

2. We split chips into two groups and do pairwise test between the two groups. If the outcome of a pairwise test is “at least one is bad”, we take away the two chips to at least eliminate one bad chips exactly. If the outcome is “both are good or both are”, we pick up one and take away the other one. We do above operations recursively until one chip is left. The last chip must be good because more than n/2 chips are good. $T(n) = T(n/2) + n/2 = \theta(n)$

3. We can find one single good chip by the method in question 2 with $\theta(n)$. After we find one good chip, we can compare it with all other chips, which takes $\theta(n)$ times. So the good chips can be identified with $\theta(n)$ pairwise tests.

problem 5 | 30.2-7. $P(x)=\prod_{j=0}^{n-1}(x-z_j)$.

Introducing devide-and-conquer strategy, $P(x)=A(x)\cdot B(x)$,

where $A(x)=\prod_{j=0}^{\lfloor \frac{n}{2} \rfloor}(x-z_j)$, $B(x)=\prod_{j=\lfloor \frac{n}{2} \rfloor+1}^{n-1}(x-z_j)$.

Using FFT to multiply two polynomials $A(x)$ and $B(x)$ of degree-bound $\frac{n}{2}$, it only takes $\Theta(\frac{n}{2}\cdot\lg\frac{n}{2}) \Leftrightarrow \Theta(n\lg n)$.

we use devide-and-conquer strategy recursively,

$$P_n(x) \qquad\qquad\qquad \frac{n}{2}\lg\frac{n}{2}$$

$$P_{n/2}(x) \qquad P_{n/2}(x) \qquad\qquad \frac{n}{4}\lg\frac{n}{4}$$

$$\cdots \qquad\qquad \cdots \qquad\qquad \vdots$$

$$(x-z_0)\ (x-z_1)\ \cdots\quad\cdots\ (x-z_{n-2})\ (x-z_{n-1}). \qquad T(1).$$

$\because$ we know the values of z_j $(j=0,1,\cdots,n-1)$, so we can get coefficients of every layer from the bottom layer.

$\therefore$ It is easy to compute point-value forms in low layers ~~every layers~~, and then use FFT. and inverse FFT. to obtain the coefficients ~~of next to~~ to new polynomials in high layers.

$\therefore$ $T(n)=2T(n/2)+n\lg n$.

apply to master theorem, case 2.

$\therefore$ $T(n)=\Theta(n\lg^2 n)$

Problem 6 30-1 Divide-and-conquer multiplication.

a. $(ax+b)(cx+d) = acx^2+adx+bcx+bd = acx^2+((a+b)(c+d)-ac-bd)x+bd$

because $(a+b)(c+d) = ac+ad+bc+bd$.

The latter expression has only three multiplications.

b. High-Low algorithm:

$$P(x) = \sum_{j=0}^{n-1} p_j x^j = \sum_{j=0}^{m-1} p_j x^j + x^m \cdot \sum_{k=m}^{n-1} p_k x^{k-m} = A \cdot x^m + B.$$

where $A = p_m + p_{m+1} \cdot x + \cdots + p_{n-1} \cdot x^{n-m-1}$

$B = p_0 + p_1 \cdot x + \cdots + p_{m-1} x^{m-1}$.

Likewise.

$$Q(x) = \sum_{j=0}^{n-1} q_j x^j = \sum_{j=0}^{m} q_j x^j + x^m \cdot \sum_{k=m}^{n-1} q_k x^{k-m} = C \cdot x^m + D.$$

where $C = q_m + q_{m+1} \cdot x + \cdots + q_{n-1} \cdot x^{n-m-1}$

$D = q_0 + q_1 \cdot x + \cdots + q_{m-1} x^{m-1}$.

Using the result of part a, we have,

$$(A \cdot x^m + B)(C \cdot x^m + D) = A \cdot C \cdot x^{2m} + ((A+B)(C+D) - AC - BD)x^m + BD \quad ①$$

Even-Odd algorithm.

$$P = \sum_{j=0}^{n-1} p_j x^j = p_0 + p_2 x^2 + \cdots + p_{2k} x^{2k} \quad (k = \lfloor \tfrac{n}{2} \rfloor, m = \lceil \tfrac{n}{2} \rceil).$$

$$+ x(p_1 + p_3 x^2 + \cdots + p_{2m-1} x^{2m-2}).$$

$$= Ax + B.$$

where $A = p_1 + p_3 x^2 + \cdots + p_{2m-1} x^{2m-2}$

$B = p_0 + p_2 x^2 + \cdots + p_{2k} x^{2k}$

$$Q = \sum_{j=0}^{n-1} q_j x^j = q_0 + q_2 x^2 + \cdots + q_{2k} x^{2k} \quad (k = \lfloor \tfrac{n}{2} \rfloor, m = \lceil \tfrac{n}{2} \rceil)$$

$$+ x(q_1 + q_3 x^2 + \cdots + q_{2m-1} x^{2m-2})$$

~~where~~ $= Cx + D$

where $C = q_1 + q_3 x^2 + \cdots + q_{2m-1} x^{2m-2}$

$D = q_0 + q_2 x^2 + \cdots + q_{2k} x^{2k}$.

$$\Rightarrow (Ax+B)(Cx+D) = A \cdot C \cdot x^2 + ((A+B)(C+D) - AC - BD)x + BD. \quad ②$$

From ① and ②, we know complexity of operations in High-Low and Even-Odd algorithms are both: $T(n) = 3T(\frac{n}{2}) + O(n)$, use the master theorem,

we get $T(n) = \Theta(n^{\lg 3})$

C. n-bit integer can be represented by. $d_{n-1}d_{n-2}\cdots d_1 d_0$, which can be represented by degree-n polynomial at $x=2$.

$$N = d_{n-1}\cdot 2^{n-1} + \cdots + d_1\cdot 2^1 + d_0 \cdot 2^0.$$

any algorithms of part b can be applied to this case.

Use High-Low algorithm:

$$P = d_0\cdot 2^0 + d_1\cdot 2^1 + \cdots + d_{k-1}\cdot 2^{k-1}$$
$$+ (d_k\cdot 2^0 + d_{k+1}\cdot 2^1 + \cdots + d_{n-1}\cdot 2^{n-k-1})\cdot 2^k$$
$$= A\cdot 2^k + B$$

where $A = d_k\cdot 2^0 + d_{k+1}\cdot 2^1 + \cdots + d_{n-1}\cdot 2^{n-k-1}$

$B = d_0\cdot 2^0 + d_1\cdot 2^1 + \cdots + d_{k-1}\cdot 2^{k-1}$

$$Q = C_0\cdot 2^0 + C_1\cdot 2^1 + \cdots + C_{k-1}\cdot 2^{k-1}$$
$$+ (C_k\cdot 2^0 + C_{k+1}\cdot 2^1 + \cdots + C_{n-1}\cdot 2^{n-k-1})\cdot 2^k$$
$$= C\cdot 2^k + D$$

where $C = C_k\cdot 2^0 + C_{k+1}\cdot 2^1 + \cdots + C_{n-1}\cdot 2^{n-k-1}$

$D = C_0\cdot 2^0 + C_1\cdot 2^1 + \cdots + C_{k-1}\cdot 2^{k-1}$

So. $(A\cdot 2^k + B)(C\cdot 2^k + D) = AC\cdot 2^{2k} + ((A+B)(C+D) - AC - BD)\cdot 2^k + BD.$

and its complexity: $T(n) = 3T(\frac{n}{2}) + O(n)$
$$= O(n^{\lg 3})$$

Problem 7

7. Example ($n=4$, complex), $w_4 = i$

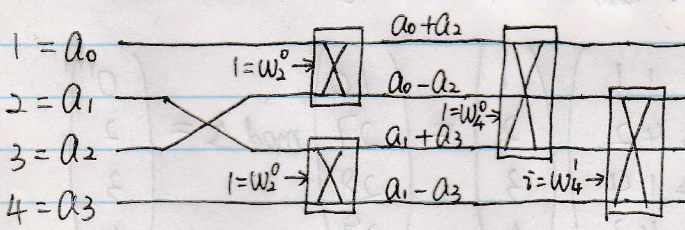
7(a).
$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & w_4 & w_4^2 & w_4^3 \\ 1 & w_4^2 & w_4^4 & w_4^6 \\ 1 & w_4^3 & w_4^6 & w_4^9 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \Rightarrow V_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

7(b) $\because$ the (j,k) entry of V_n^{-1} is w_n^{-kj}/n, and $\frac{1}{i} = -i$

$\therefore$
$$V_4^{-1} = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & -i/4 & -1/4 & i/4 \\ 1/4 & -1/4 & 1/4 & -1/4 \\ 1/4 & i/4 & -1/4 & -i/4 \end{pmatrix} = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & -\frac{i}{4} & -\frac{1}{4} & \frac{i}{4} \\ \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & \frac{i}{4} & -\frac{1}{4} & -\frac{i}{4} \end{pmatrix}$$

7(c) $\vec{y} = V_4 \vec{a} \Rightarrow$
$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 10 \\ -2-2i \\ -2 \\ -2+2i \end{pmatrix}$$

7(d)



$y_0 = a_0 + a_2 + a_1 + a_3 = 10$

$y_1 = a_0 - a_2 + (a_1 - a_3)\cdot i = -2-2i$

$y_2 = a_0 + a_2 - a_1 - a_3 = -2$

$y_3 = a_0 - a_2 - \cancel{a_1+a_3}(a_1 - a_3)\cdot i = -2+2i$

Problem 8

8. Example $(n=4, \bmod 5)$. $w_4 = 3$, $\Rightarrow w_4^2 = 9 \bmod 5 = 4$, $w_4^3 = 27 \bmod 5 = 2$.

8(a)
$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & w_4 & w_4^2 & w_4^3 \\ 1 & w_4^2 & w_4^4 & w_4^6 \\ 1 & w_4^3 & w_4^6 & w_4^9 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \Rightarrow V_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 4 & 2 \\ 1 & 4 & 1 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix}$$

8(b) the (j,k) entry of V_n^{-1} is w_n^{-kj}/n.

$$V^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{3} & \frac{1}{4} & \frac{1}{2} \\ 1 & \frac{1}{4} & 1 & \frac{1}{4} \\ 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{3} \end{pmatrix} \cdot \frac{1}{4}$$, in order to make the entry of V^{-1} be integer,

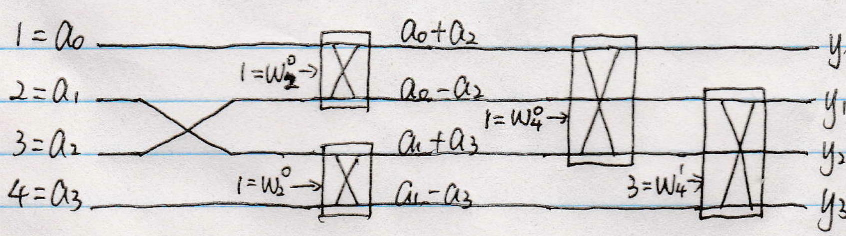
let
$$V_4^{-1} = 3 \cdot 48 \cdot V^{-1} = \begin{pmatrix} 144 & 144 & 144 & 144 \\ 144 & 48 & 36 & 72 \\ 144 & 36 & 144 & 36 \\ 144 & 72 & 36 & 48 \end{pmatrix} \bmod 5 = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 4 & 3 & 1 & 2 \\ 4 & 1 & 4 & 1 \\ 4 & 2 & 1 & 3 \end{pmatrix}$$

In this case, $V_4^{-1} \cdot V_4 \bmod 5 = I$

because $[V_4^{-1} \cdot V_n]_{jk} = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$.

8(c) $\vec{y} = V_4 \vec{a} \Rightarrow$
$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 4 & 2 \\ 1 & 4 & 1 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 10 \\ 27 \\ 28 \\ 29 \end{pmatrix} \bmod 5 = \begin{pmatrix} 0 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

8(d)



$y_0 = a_0 + a_2 + a_1 + a_3 = 10 \bmod 5 = 0$

$y_1 = a_0 - a_2 + (a_1 - a_3) \cdot 3 \quad = 2 \quad$ ①

$y_2 = a_0 + a_2 - a_1 - a_3 \quad = 3 \quad$ ②

$y_3 = a_0 - a_2 - (a_1 - a_3) \cdot 3 \quad = 4 \quad$ ③

① $= a_0 - a_2 + (a_1 - a_3) \cdot 3 = 1 - 3 + (2-4) \cdot 3 = (5+1) - 3 + (5+2-4) \cdot 3 = 12 \bmod 5 = 2$

② $= a_0 + a_2 - a_1 - a_3 = 4 - 6 = 5 + 4 - 6 = 3$

③ $= a_0 - a_2 - (a_1 - a_3) \cdot 3 = (5+1) - 3 - (5+2-4) \cdot 3 = 3 - 9 = 5 \times 2 + 3 - 9 = 4$.