Asked : find the optimum assignment for minimizing cost ??
by using **Hungarian Algorithms**

Given :

$$[A]_{Given} = \begin{array}{c|c|c|c|c|c} & 1 & 2 & 3 & 4 & 5 \\ 1 & 5 & 2 & 4 & 3 & 6 \\ 2 & 7 & 4 & 3 & 6 & 5 \\ 3 & 2 & 4 & 6 & 8 & 7 \\ 4 & 8 & 6 & 3 & 5 & 4 \\ 5 & 3 & 9 & 4 & 7 & 6 \end{array} = \text{Job}\#$$

Person#

~~ITemp01~~ ~~Irowmin~~ (1) = 2
(2) = 3
(3) = 2
(4) = 3
(5) = 3

objAdded = 2+3+2+3+3 = 13

1/5

STEP1. subtract each row with its corresponding minimum value

$$[A_1] = \begin{matrix} 3 & 0 & 2 & 1 & 4 \\ 4 & 1 & 0 & 3 & 2 \\ 0 & 2 & 4 & 6 & 5 \\ 5 & 3 & 0 & 2 & 1 \\ 0 & 6 & 1 & 4 & 3 \\ \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \end{matrix}$$

A

$$\begin{matrix}\text{ITcolmin} \\ \text{ITEMP01}\end{matrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{Bmatrix} \Rightarrow \text{objadded} = \textcircled{13} + 1 + 1 = 15$$

step2 : subtract each column with its corresponding minimum value

$$[A_2] = \begin{bmatrix} 3 & 0 & 2 & 0 & 3 \\ 4 & 1 & 0 & 2 & 1 \\ 0 & 2 & 4 & 5 & 4 \\ 5 & 3 & 0 & 1 & 0 \\ 0 & 6 & 1 & 3 & 2 \end{bmatrix} \rightarrow \begin{matrix}\text{NUMZPerRow} \\ \text{ITEMP01}\end{matrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{Bmatrix} 2 \\ 1 \\ 1 \\ 2 \\ 1 \end{Bmatrix}$$

$$\text{NUMZPerCol} \begin{pmatrix} \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \end{pmatrix} = \begin{Bmatrix} 2 \\ 1 \\ 2 \\ 1 \end{Bmatrix}$$

Step3 : Cover all zeros with min. # of lines

$$[A_2] = \begin{bmatrix} 3 & 0 & 2 & 0 & 3 \\ 4 & 1 & 0 & 2 & 1 \\ 0 & 2 & 4 & 5 & 4 \\ 5 & 3 & 0 & 1 & 0 \\ 0 & 6 & \boxed{1} & 3 & 2 \end{bmatrix} \begin{matrix} \rightarrow \text{First} \\ \rightarrow \text{FOURTH} \\ \\ \rightarrow \text{Second} \\ \\ \end{matrix}$$

THIRD

$$\Rightarrow \text{ITEMP01} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{Bmatrix} 0 \\ 1 \rightarrow 0 \\ \cancel{1} \rightarrow 0 \\ \cancel{2} \rightarrow 0 \\ 1 \rightarrow 0 \end{Bmatrix}$$

$$\text{ITEMP02} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{Bmatrix} 2 \rightarrow 0 \\ 0 \\ 2 \rightarrow 1 \rightarrow 0 \\ 0 \\ 1 \rightarrow 0 \end{Bmatrix}$$

cover numbers pass through { rows # 1, 4, 2 ⇒ m = # uncovered rows = 2
{ col # 1 ⇒ r = # uncovered cols = 4

since # of Minimum lines = 4 (= 3 rows + 1 col) < N = 5, hence

Min. uncover number = 1 = p

$\boxed{2/5}$

Hence: objadded $= \underset{\text{previous value}}{\textcircled{15}} + p(m + r - n) = 16$

$1*(2+4-5)$

$$[A_3] = \begin{array}{c|ccccc|l}
 & \text{THIRD} & & \text{FIRST} & & & \\
\text{SECOND} & 4 & 0 & 2 & 0 & 3 & \\
 & 5 & 1 & 0 & 2 & 1 & \\
 & 0 & 1 & 3 & 4 & 3 & \\
 & 4 & 3 & 0 & 1 & 0 & \text{FOURTH} \\
 & 0 & 5 & 0 & 2 & 1 &
\end{array}$$

Itempo1 $\begin{pmatrix}1\\2\\3\\4\\5\end{pmatrix} = \begin{cases}2 \to 0\\ 1 \to 0\\ 1 \to 0\\ 2 \to 1\\ 2 \to 1 \to 0\end{cases}$

Itempo2 $\begin{pmatrix}1\\2\\3\\4\\5\end{pmatrix} = \begin{cases}2 \to 0\\ 1 \to 0\\ 3 \to 0\\ 1 \to 0\\ 1\end{cases}$

step 3 again

cover numbers pass through { rows # 1, 4 ⇒ m = 3 = # uncover rows
{ cols # 3, 1 ⇒ r = 3 = # uncover cols

$p = 1$

objadded $= \textcircled{16} + 1(3+3-5) = \textcircled{17}$ optimum cost (see step #4)

step 4

$$[A_4] = \begin{array}{c|ccccc|l}
\text{THIRD} & 5 & 0 & 3 & \overset{2^{nd}}{(0)} & 3 & \to 2 \text{ zeros} \to 0 \\
\text{First} & 5 & \overset{3^{rd}\text{ditto}}{(0)} & 0 & 1 & 0 & \to 3 \to 0 \\
\text{Fourth} & \overset{4^{th}}{(0)} & 0 & 3 & 3 & 2 & \to 2 \to 0 \\
\text{Fifth} & 7 & 3 & 1 & 1 & \overset{1^{st}}{(0)} & \to 1 \\
\text{SECOND} & 0 & 4 & \overset{5^{th}}{(0)} & 1 & 0 & \to 3 \to 0 \\
\hline
 & 2 \text{ zeros} & 3 & 3 & 1 & 3 \\
 & 1 & 2 & 1 & 0 & 2 \\
 & 0 & 1 & 0 & & 1 \\
 & & 0 & & & 0
\end{array}$$

cover numbers passed through { rows # 2, 5, 1, 3, 4

Since # of min. lines = 5 = N ⇒ STOP process and ready to ASSIGN Jobs

Conclusions

			Based on Original Matrix [A]
4th person	assigned to 5th job	⇒	cost = 4 units
1st "	4th	⇒	cost = 3
2nd "	2nd	⇒	cost = 4
3rd "	1st	⇒	cost = 2
5th	3rd	⇒	cost = 4
			Total optimum Cost = 17

3/5

ASSIGNMENT PROBLEMS

The assignment problems arise in manufacturing industries and businesses. The general class of problems are similar to the assignment of n jobs to n machines in a production operation. Other problems may be like assignment of teachers to courses, workers to jobs, pilots to flights, and so on. There is cost c_{ij} associated with job i assigned to machine j. **c** is referred to as the *effectiveness matrix*. The corresponding variable x_{ij} takes the value of 1. The general problem can be put in the form:

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\ &\text{subject to} && \sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, 2, \ldots, n \\ &&& \sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, 2, \ldots, n \\ &&& x_{ij} = 0 \quad \text{or} \quad 1 \quad i = 1, 2, \ldots, n; \quad j = 1, 2, \ldots, n \end{aligned} \tag{10.8}$$

We may view the problem as a special case of the transportation problem. If the number of jobs is not equal to the number of machines, dummy jobs or machines may be added as needed to balance the problem. The associated costs may be left as zeros. The requirement that each variable be an integer equal to 0 or 1 is automatically satisfied by the triangular property of the basis. The problem can be solved using the transportation algorithm. However, the assignment problem has a special structure which enables us to solve by simpler manipulations. We present here the method originally developed by two Hungarian mathematicians.

Ref. "Optimization Concepts and Applications in Engineering"; by Belegundu & Chandrupatla; Prentice-Hall (1999); pages 347-348

The Hungarian Method

In developing the Hungarian method for the solution of the assignment problem, we develop two preliminary results.

Effect of Subtracting p_i from row i and q_j from column j. If we define the new effectiveness coefficient as $c'_{ij} = c_{ij} - p_i - q_j$, and consider the objective function with these coefficients, then

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c'_{ij}x_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij} - \sum_{i=1}^{n} p_i \sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} q_j \sum_{i=1}^{n} x_{ij}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij} - \sum_{i=1}^{n} p_i - \sum_{j=1}^{n} q_j \qquad (10.9)$$

The objective function is reduced by the sum of the row and column subtractions. The minimum point does not change, but the value gets reduced by the sum of values subtracted. This sum must be added to the new optimum value to get the original optimum. The strategy is to first subtract the row minimum from each row successively, and then the column minimum from each column successively. Then each row and each column has at least one zero in it. If we can perform the assignments at the zero locations, the minimum of the new objective is zero. If assignment is not possible, additional manipulations are necessary. One other manipulation involves partitioned matrices of **c**.

Manipulation of a Partition of c. We develop an identity for the indices defined as.

$$0 = p \sum_{i=1}^{m} \sum_{j=1}^{r} x_{ij} + p \sum_{i=1}^{m} \sum_{j=1}^{r} x_{ij} = p \sum_{i=1}^{m} \sum_{j=1}^{r} x_{ij} + p \sum_{i=1}^{m} \left(1 - \sum_{j=r+1}^{n} x_{ij}\right)$$

$$= -p \sum_{i=1}^{m} \sum_{j=1}^{r} x_{ij} + mp - p \sum_{i=1}^{m} \sum_{j=r+1}^{n} x_{ij} = mp - p \sum_{i=1}^{m} \sum_{j=1}^{r} x_{ij} - p \sum_{j=r+1}^{n} \left(1 - \sum_{i=m+1}^{n} x_{ij}\right)$$

$$= p(m - r - n) - p \sum_{i=1}^{m} \sum_{j=1}^{r} x_{ij} - p \sum_{i=m+1}^{n} \sum_{j=r+1}^{n} x_{ij} \qquad (10.10)$$

The result of this evaluation is the identity

$$p \sum_{i=1}^{m} \sum_{j=1}^{r} x_{ij} + p \sum_{i=m+1}^{n} \sum_{j=r+1}^{n} x_{ij} = -p(m + r - n) \qquad (10.11)$$

Eq. (10.11) can be interpreted as follow. Subtracting p from every element from the partition c_{ij}, $i = 1$ to m, $j = 1$ to r, and adding p to each of the elements of the partition c_{ij}, $i = m + 1$ to n, $j = r + 1$ to n, reduces the objective function by the constant value $p(m + r - n)$. If such a manipulation is done, this constant quantity must be added to the objective function value to get the value of the original problem.

With the concepts of the basic manipulations fixed, we present the steps in the assignment calculations.

Remarks

[1] The above basic Hungarian Algorithms can be (slightly) modified to handle "MAX. obj", and/or for the case where the given coefficient matrix is a "Rectangular matrix" (instead of a "square matrix")

[2] Assuming NP processors are available, what will be the "best parallel computation strategies" to solve large-scale problems (in terms of computer memory & computational time requirements) ??

[3] The Hungarian Algorithm's output of the above 5×5 example can be expressed in the following form:

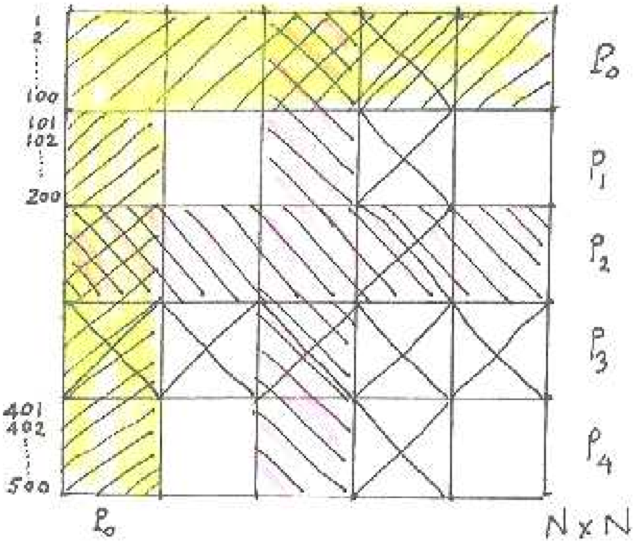
$$IPERM \underset{\uparrow \text{Person}^{\#}}{\begin{pmatrix}1\\2\\3\\4\\5\end{pmatrix}} = \underset{\uparrow \text{Job}^{\#}}{\begin{Bmatrix}4\\2\\1\\5\\3\end{Bmatrix}}$$

Thus, can we use/apply "Hungarian Algo." to find IPERM(new# system) = {old# system}, example: similar to METis, ND, MMD reordering algorithms to minimize Fill-in Terms from Sparse Symbolic Factorization ??

[4] Based on Belegundu's implementation of Hungarian algorithms, we do NOT need to copy the "original, big" coefficient matrix, since the objective function value can be continuously updated !

Parallel "Hungarian" Strategies

(a) Storage Schemes/strategies



Say $N = 500$
$NP = 5$ processors

Hence, each processor stores "block" rows (= 100 rows) and "block" columns (= 100 cols)

Example:

P_0 stores rows $i = 1 \to 100$; col $j = 1 \to 500$
rows $j = 1 \to 500$; cols $i = 1 \to 100$

P_0 = Yellow area ▨

P_2 stores rows $i = 201 \to 300$; col $j = 1 \to 500$
rows $j = 1 \to 500$; col $i = 201 \to 300$

P_2 = Pink area ▧

(b) In general, if NP processors are used, then each processor stores according to the following $\text{RATIO} = \frac{2(NP)-1}{(NP)^2}$ of the total/original matrix space

If $N = 500$ & $NP = 5$

Then $\text{Ratio} = \frac{2(5)-1}{(5)^2} = \frac{9 \text{ blocks}}{25 \text{ blocks}} = 0.36 \approx 36\%$ of the original/total matrix space

If NP = Large, Then $\text{Ratio} = \frac{2(NP)-1}{(NP)^2} \approx \frac{2\cancel{(NP)}}{(NP)^{\cancel{2}}} \approx \frac{2}{NP} \approx$ small fraction of the original (large) matrix

---

If $N = 503$ & $NP = 5$, Then $\text{Extra} = \text{Mod}(503, 5) = 3 \Rightarrow$

P_0 stores 100 + 1 rows
P_1 stores 100 + 1 rows
P_2 " 100 + 1 rows
P_3 " 100 rows
P_4 " 100 rows

```
If [ME.LT. Extra] Then
    IAverows = $\frac{N}{NP} + 1$
Else
    Iaverows = $\frac{N}{NP}$
Endif
```

(c) Parallel Hungarian Algo.

INPUT PHASE

- Each processor READ (or GENERATE) block of rows
  100% parallel ?
- Each processor SENDS appropriated submatrices to other processors

  Each processor RECEIVES appropriated submatrices from other processors

TASK 1 = subtract each row with its own minimum row value

- Each processor performs Task 1 {thus, original matrix [A] will be changed (updated)}
- Each processor sends to (and receive from) appropriated/updated matrices other processors

TASK 2 = subtract each column with its own minimum col. value

- Each processor performs Task 2 {thus, original matrix [A] will be further changed/updated}
- Each processor sends to (and receive from) appropriated/updated matrices other processors

TASK 3 = Covered all zeros with Minimum # lines (= Nlines)

```
If [Nlines .LT. N] Then
    update the matrix and Go back To Task 3
Elseif [NLINES .EQ. N] Then
    GO TO Task 4
Endif
```

Task 4 = Make Assignments, print optimum objective function value (Minimum) then QUIT

The Hungarian Algorithm

Assumption: There are n "jobs" and n "machines".

Step 0: If necessary, convert the problem from a *maximum* assignment into a *minimum* assignment. We do this by letting C = maximum value in the assignment matrix. Replace each c_{ij} with $C - c_{ij}$. (see example 2).

Step1: From each row subtract off the row min.

Step 2: From each column subtract off the row column min.

Step 3: Use as few lines as possible to cover all the zeros in the matrix. There is no easy rule to do this – basically trial and error.
Suppose you use k lines.

- If $k < n$, let m be the minimum *uncovered* number. Subtract m from every uncovered number. Add m to every number covered with **two** lines. Go back to the start of step 3.
- If $k = n$, goto step 4.

Step 4: Starting with the top row, work your way downwards as you make assignments. An assignment can be (uniquely) made when there is exactly one zero in a row. Once an assignment it made, delete that row and column from the matrix.

If you cannot make all n assignments and all the remaining rows contain more than one zero, switch to columns. Starting with the left column, work your way rightwards as you make assignments.

Iterate between row assignments and column assignments until you've made as many unique assignments as possible. If still haven't made n assignments and you cannot make a unique assignment either with rows or columns, make one arbitrarily by selecting a cell with a zero in it. Then try to make unique row and/or column assignments. (See the examples below).

Example 1

The Police Department of Fargo, ND begins the morning shift by informing all first-shift patrol persons of the previous evening's activities and giving assignments to the various team members. On a particular day, the following three activites must be accomplished: (1) delivery of a DARE (Drug Abuse Resistance Education) lecture at a local elementary school; (2) instruction of the rookie ploice class in using the baton; and (3) preparation of a report for the evening's city council meeting on drug activities over the past three months.

To help fulfill the mayor's promise of "keeping more police on the streets," only three officers will be assigned to these activities. The goal is to minimize the total time these officers will be absent from street patrol. Given the expertise of each officer, the table below shows the time estimates (in hours) for each activity, including any officer specific preparation time. Because the assignments will be occurring simultaneously, each officer will be assigned to one and only one of the activities.

Officer	*DARE Lecture*	*Baton Training*	*Drug Analysis*
Borel	4	2	8
Frank	4	3	7
Klaus	3	1	6

Solve using the Hungarian Algorithm.

Step 0: The problem is already in minimization form.

Officer	*DARE Lecture*	*Baton Training*	*Drug Analysis*	*Row Mins*
Borel	4	2	8	2
Frank	4	3	7	3
Klaus	3	1	6	1

Step 1: Subtract the row mins from each row.

Officer	*DARE Lecture*	*Baton Training*	*Drug Analysis*
Borel	2	0	6
Frank	1	0	4
Klaus	2	0	5
Col Mins	1	0	4

Step 2: Subtract the column mins from each column.

Officer	*DARE Lecture*	*Baton Training*	*Drug Analysis*
Borel	1	0	2
Frank	0	0	0
Klaus	1	0	1

Step 3: Use the minimum number of lines to cover all the zeros in the matrix. Put lines through column 2 and row 2. Since the number of lines required is $k = 2 < n = 3$, we set m = minimum uncovered number. So $m = 1$. We subtract m from every uncovered number and add m to every number covered with two lines.

Officer	*DARE Lecture*	*Baton Training*	*Drug Analysis*
Borel	0	0	1
Frank	0	1	0
Klaus	0	0	0

Step 3: Use the minimum number of lines to cover all the zeros in the matrix. The number of lines required is $k = n = 3$, so we STOP.

Step 4: We can make no unique assignments with rows or columns. So we start with an aribitrary assignment where a 0 appears in a cell. Assign Borel to DARE and delete this row and column from the matrix. We are left with the following submatrix.

Officer	*Baton Training*	*Drug Analysis*
Frank	1	0
Klaus	0	0

Assign Frank to Drug Analysis and Klaus to Baton Training.

What if we had assigned Borel to Baton Training instead of DARE? Then, we would have been left with the following submatrix.

Officer	*DARE Lecture*	*Drug Analysis*
Frank	0	0
Klaus	0	0

We can make assignments arbitrarily at this point. So there are 3 optimal solutions. (What are the other two?)

Example 2

The accounting firm of Barnes, Fernandez, and Chou (BFC) has just hired six new junior accountants who are to be placed into six specialty areas within the firm. Each applicant has been given an overall skills test in the specialty areas. The results are presented in the table below.

Junior Accountant	Auditing	Corporate Tax	Personal Tax	Financial Analysis	Information Systems	General Accounting
Amy Cheng	62	75	80	93	95	97
Bob Szary	75	80	82	85	71	97
Sue Crane	80	75	81	98	90	97
Maya Pena	78	82	84	80	50	98
Koo Thanh	90	85	85	80	85	99
Lyn Ortiz	65	75	80	75	68	96

a. If test scores are judged to be a measure of potential success, which junior accountant should be assigned to which specialty?

We can formulate this as an maximum assignment problem. Observe that if we try to just assign the highest scores we end up with

Junior Accountant	Specialty Area	Score
Amy Cheng	Information Systems	95
Bob Szary	Corporate Tax	80
Sue Crane	Financial Analysis	98
Maya Pena	Personal Tax	84
Koo Thanh	General Accounting	99
Lyn Ortiz	Auditing	65
	Total Score	521

Can we do better? The Hungarian Algorithm will give the optimal assignment.

Step 0: Since this is a **maximum** assignment, we must convert to a **minimum** assignment by replacing each cell entry c_{ij} with $C - c_{ij}$ where $C =$ maximum cell value. Here $C = 99$. We now have a matrix that we can run the Hungarian algorithm on.

	Auditing	Corporate Tax	Personal Tax	Fin Anal	Info Sys	Gen Acct	Row Mins
Amy	37	24	19	6	4	2	2
Bob	24	19	17	14	28	2	2
Sue	19	24	18	1	9	2	1
Maya	21	17	15	19	49	1	1
Koo	9	14	14	19	14	0	0
Lyn	34	24	19	24	31	3	3

Step 1: Subtract the row mins from each row.

	Auditing	Corporate Tax	Personal Tax	Fin Anal	Info Sys	Gen Acct
Amy	35	22	17	4	2	0
Bob	22	17	15	12	26	0
Sue	18	23	17	0	8	1
Maya	20	16	14	18	48	0
Koo	9	14	14	19	14	0
Lyn	31	21	16	21	28	0
Col Mins	9	14	14	0	2	0

Step 2: Subtract the column mins from each column.

	Auditing	Corporate Tax	Personal Tax	Fin Anal	Info Sys	Gen Acct
Amy	26	8	3	4	0	0
Bob	13	3	1	12	24	0
Sue	9	9	3	0	6	1
Maya	11	2	0	18	46	0
Koo	0	0	0	19	12	0
Lyn	22	7	2	21	26	0

Step 3: Use the minimum number of lines to cover all the zeros in the matrix. Put lines through columns 3, 4, 5, and 6 and row 5. Since the number of lines required is $k = 5 < n = 6$, we set m = minimum uncovered number. So $m = 2$. We subtract m from every uncovered number and add m to every number covered with two lines.

	Auditing	Corporate Tax	Personal Tax	Fin Anal	Info Sys	Gen Acct
Amy	24	6	3	4	0	0
Bob	11	1	1	12	24	0
Sue	7	7	3	0	6	1
Maya	9	0	0	18	46	0
Koo	0	0	2	21	14	2
Lyn	20	5	2	21	26	0

Step 3: Use the minimum number of lines to cover all the zeros in the matrix. Put lines through columns 4, 5, and 6 and rows 4 and 5. Since the number of lines required is $k = 5 < n = 6$, we set m = minimum uncovered number. So $m = 1$. We subtract m from every uncovered number and add m to every number covered with two lines.

	Auditing	Corporate Tax	Personal Tax	Fin Anal	Info Sys	Gen Acct
Amy	23	5	2	4	0	0
Bob	10	0	0	12	24	0
Sue	6	6	2	0	6	1
Maya	9	0	0	19	47	1
Koo	0	0	2	22	15	3
Lyn	19	4	1	21	26	0

Step 3: Use the minimum number of lines to cover all the zeros in the matrix. The number of lines required is $k = n = 6$, so we STOP.

Step 4: Make the assignment as follows: Starting with the top row going down, we can assign Sue to Financial Analysis and Lyn to General Accounting.

	Auditing	Corporate Tax	Personal Tax	Info Sys
Amy	23	5	2	0
Bob	10	0	0	24
Maya	9	0	0	47
Koo	0	0	2	15

Going back to the top row, we assign Amy to Info Systems. The remaining rows have two or more zeros in them so we switch to columns. Starting from the left column, we assign Koo to Auditing.

	Corporate Tax	Personal Tax
Bob	0	0
Maya	0	0

We are left with a zero matrix which means we can make arbitrary assignments. Assign Bob to Corporate Tax and Maya to Personal Tax. The total score of this assignment is 543.

b. In addition to the primary specialty area, each new junior accountant will be trained in a second specialty so the company has two junior accountants with training in each specialty. Eliminating the original assignments found in part a, what assignments now give the maximal potential for the second specialty?

We'd like to use the Hungarian algorithm but to do so, we need to first eliminate the original assignments from the cost matrix. We do this by putting a very low potential (i.e., zero) to these assignments.

Junior Accountant	Auditing	Corporate Tax	Personal Tax	Financial Analysis	Information Systems	General Accounting
Amy Cheng	62	75	80	93	0	97
Bob Szary	75	0	82	85	71	97
Sue Crane	80	75	81	0	90	97
Maya Pena	78	82	0	80	50	98
Koo Thanh	0	85	85	80	85	99
Lyn Ortiz	65	75	80	75	68	0

Now we proceed as we did before.

Step 0: Since this is a **maximum** assignment, we must convert to a **minimum** assignment by replacing each cell entry c_{ij} with $C - c_{ij}$ where $C =$ maximum cell value. Here $C = 99$. We now have a matrix that we can run the Hungarian algorithm on.

	Auditing	Corporate Tax	Personal Tax	Fin Anal	Info Sys	Gen Acct	Row Mins
Amy	37	24	19	6	99	2	2
Bob	24	99	17	14	28	2	2
Sue	19	24	18	99	9	2	2
Maya	21	17	99	19	49	1	1
Koo	99	14	14	19	14	0	0
Lyn	34	24	19	24	31	99	19

Step 1: Subtract the row mins from each row.

	Auditing	Corporate Tax	Personal Tax	Fin Anal	Info Sys	Gen Acct
Amy	35	22	17	4	97	0
Bob	22	97	15	12	26	0
Sue	17	22	16	97	7	0
Maya	20	16	98	18	48	0
Koo	99	14	14	19	14	0
Lyn	15	5	0	5	12	80
Col Mins	15	5	0	4	7	0

Step 2: Subtract the column mins from each column.

	Auditing	Corporate Tax	Personal Tax	Fin Anal	Info Sys	Gen Acct
Amy	20	17	17	0	90	0
Bob	7	92	15	8	19	0
Sue	2	17	16	93	0	0
Maya	5	11	98	14	41	0
Koo	84	9	14	15	7	0
Lyn	0	0	0	1	5	80

Step 3: Use the minimum number of lines to cover all the zeros in the matrix. Put lines through columns 4, 5, and 6 and row 6. Since the number of lines required is $k = 4 < n = 6$, we set $m =$ minimum uncovered number. So $m = 2$. We subtract m from every uncovered number and add m to every number covered with two lines.

	Auditing	Corporate Tax	Personal Tax	Fin Anal	Info Sys	Gen Acct
Amy	18	15	15	0	90	0
Bob	5	90	13	8	19	0
Sue	0	15	14	93	0	0
Maya	3	9	96	14	41	0
Koo	82	7	12	15	7	0
Lyn	0	0	0	3	7	82

Step 3: Use the minimum number of lines to cover all the zeros in the matrix. Put lines through columns 4 and 6 and rows 3 and 6. Since the number of lines required is $k = 4 < n = 6$, we set $m =$ minimum uncovered number. So $m = 3$. We subtract m from every uncovered number and add m to every number covered with two lines.

	Auditing	Corporate Tax	Personal Tax	Fin Anal	Info Sys	Gen Acct
Amy	15	12	12	0	87	0
Bob	2	87	10	8	16	0
Sue	0	15	14	96	0	3
Maya	0	6	93	14	38	0
Koo	79	4	9	15	4	0
Lyn	0	0	0	6	7	85

Step 3: Use the minimum number of lines to cover all the zeros in the matrix. Put lines through columns 1, 4, 5, and 6 and row 6. Since the number of lines required is $k = 5 < n = 6$, we set $m =$ minimum uncovered number. So $m = 4$. We subtract m from every uncovered number and add m to every number covered with two lines.

	Auditing	Corporate Tax	Personal Tax	Fin Anal	Info Sys	Gen Acct
Amy	15	8	8	0	87	0
Bob	2	83	6	8	16	0
Sue	0	11	10	96	0	3
Maya	0	2	89	14	38	0
Koo	79	0	5	15	4	0
Lyn	4	0	0	10	11	89

Step 3: Use the minimum number of lines to cover all the zeros in the matrix. The number of lines required is $k = n = 6$, so we STOP.

Step 4: Make the assignment as follows: Starting with the top row going down, we can assign Bob to General Accounting, Maya to Auditing , Koo to Corporate Tax, Lyn to Personal Tax.

	Fin Anal	Info Sys
Amy	0	87
Sue	96	0

Going back to the top row, we can assign Amy to Financial Analysis and Sue to Information Systems. The total potential of this assignment is 523.