**The Assignment Problem, continuation**

First we will consider some greedy slave algorithms, to find many independent zeros and/or cover the zeros. One can remember, that the Hungarian Method does not tell us that "how" to cover the zeros with few lines. Often we can manage this task by some greedy algorithms.

What does **GREEDY** mean (generally)?

- A greedy algorithm is always fast.
- There is no guarantee that it finds an optimal solution (but we do not mind it). The greedy strategy does not promise the optimality at all.
- Usually the following happens: We divide the decision problem to steps. In many cases there is a natural way to distinguish the steps that come in an order, after each other. Then, step by step we make locally optimal decisions: No matter what will happen later, we make a decision that is optimal for this local step (and it can be that the consequences of the current decision will not be very good later).
- Example1: Determining the rank of a matrix. We make basis transformations. We choose a column that can be chosen to enter the basis. It is true that any choice is an optimal choice: If we can take k columns into the basis, it is no matter what columns are chosen and in what order.
- Example2: minimum weight spanning tree, determining it by the Kruskal algorithm: At any step we choose the cheepest edge that will not cause cycle and add to the tree.
- Note, that greedy fails many times, i.e. it provides quite bad solutions. But we do not mind it as it consumes only a short time. If greedy fails, we try some more clever algorithm (that needs more time).

**Greedy1**. If all the zeros are in few rows and columns, we can easily cover them with no problem. Like below. (In other words: Let us choose a column or row with maximum number of zeros which are not covered yet. Cover this line, and repeat this step.)

| | | | | | |
|---|---|---|---|---|---|
| ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ |
| 0 | 2 | 3 | 5 | 4 | 7 |
| 0 | 1 | 1 | 3 | 4 | 5 |
| ~~0~~ | ~~0~~ | ~~2~~ | ~~0~~ | ~~0~~ | ~~0~~ |
| 0 | 3 | 4 | 7 | 6 | 8 |
| 0 | 1 | 2 | 4 | 4 | 4 |

**Greedy2**. Let us suppose that we want to cover all zeros by small number of lines, and also (at the same time) we would like to find many independent zeros. Supposing that there are n independent zeros, and there is a row (or a column) where there is only one zero, then we *must* choose this one (by this choice we cannot make a wrong decision!) We apply this idea: Let us choose the zero in consideration, exclude (in mind) its row and column, and continue the process in the reduced table.

Like here:

| | | | | | |
|---|---|---|---|---|---|
| **0** | 1 | 4 | 6 | 8 | 3 |
| 1 | 2 | **0** | 0 | 0 | 7 |
| 0 | 1 | 1 | **0** | 4 | 5 |
| ~~0~~ | ~~**0**~~ | ~~2~~ | ~~0~~ | ~~0~~ | ~~0~~ |
| 0 | 3 | 4 | 7 | **0** | 8 |
| 0 | 1 | 2 | 4 | 4 | 4 |

- There is only a unique zero in the first row. We choose it. At the same time, we cover its column (because it is better to choose the column than the row as there are also other zeros in the column).
- There is a unique zero in the second column. We choose it and cover its row.
- There is a unique zero in the third row (as the other one is already excluded). We choose it and cover its column.
- There is a unique zero in the fifth row (as the other one is already excluded). We choose it and cover its column.
- There is a unique zero in the second row (as all other are already excluded). We choose it and cover its column (or row).

At this moment there are not more independent zeros (as it should be in position (6,6) but it is not zero). Still, we found 5 independent zeros, and already also covered them.

The above tricks (greedy algorithms) do not work, if there are zeros “here and there”, not too many but not too few. And (after finding some “good” zeros), we do have at least two zeros in each row and each column. Then still we can make **brute force**, or **backtracking**. Both are time-consuming.

**Greedy3**. A third greedy algorithm (that can work also if Greedy1 or Greedy2 fail):

- Let us choose the first zero in the first row.
- Let us choose the first appropriate zero in the second row (if any).
- Let us choose the first appropriate zero in the third row (if any).
- Etc.

It can find (in a good case) many independent zeros, very fast.

**The Kőnig algorithm** (this is not greedy!)

Let us suppose we already do have an independent system of zeros (called **chosen zeros**), like below.

**Initially:**

- we number the rows (no need to number the columns)
- we put a + before each row where we do not have a “chosen” zero
- we put a ↓ above each column where we do not have a “chosen” zero, these columns are called “desired columns”.

| | | | | | ↓ | ↓ |
|---|---|---|---|---|---|---|
| 1 | **0** | 2 | 0 | 6 | 8 | 1 |
| 2 | 1 | 2 | 2 | **0** | 4 | 7 |
| 3 | 3 | 0 | **0** | 3 | 4 | 5 |
| 4 | 0 | **0** | 1 | 0 | 0 | 0 |
| 5 + | 0 | 3 | 0 | 7 | 6 | 4 |
| 6 + | 0 | 1 | 1 | 4 | 4 | 4 |

Then:

- We look for a +. If we find a +, we consider **all zeros** in its row, and write the number of the row of the zero above the column of the zero. Then we change the + of the row to *.
- Or, we look for a column where we already wrote a number above the column (and it is without *). Then consider the **chosen zero** in this column and put a + before the row. Then add a * to the column.
- If we arrive to some desired column, we stop.
- If there is no more + neither a column with a number but without a *, we stop.

The meaning:
+ :  We arrived to this row
* :  We already investigated this row (what columns can be reached directly from this row)
* above the columns: we already dealt with this column

Note: We can apply any kind of search, like depth-first or width-first or any other.

**Theorem**:

- If the collection of the chosen zeros is a maximal independent system of $k<n$ zeros, then the Kőnig algorithm finds k covering lines that cover all zeros.
- Otherwise (the collection of the chosen k zeros is not a maximal independent set), then the Kőnig algorithm finds k+1 independent zeros.

Proof: By construction.

These are (e.g.) the steps

1. We choose (e.g.) the fifth row, find two zeros in this row, write “5” above the columns of the zeros, and change + to * in the fifth row:

| | 5 | | 5 | | ↓ | ↓ |
|---|---|---|---|---|---|---|
| 1 | **0** | 2 | 0 | 6 | 8 | 1 |
| 2 | 1 | 2 | 2 | **0** | 4 | 7 |
| 3 | 3 | 0 | **0** | 3 | 4 | 5 |
| 4 | 0 | **0** | 1 | 0 | 0 | 0 |
| 5 * | 0 | 3 | 0 | 7 | 6 | 4 |
| 6 + | 0 | 1 | 1 | 4 | 4 | 4 |

2. Then we find the first column. In this column the chosen zero is in the first row. We put a + before the first row, and we add a * to the first column:

| | 5* | | 5 | | ↓ | ↓ |
|---|---|---|---|---|---|---|
| 1+ | **0** | 2 | 0 | 6 | 8 | 1 |
| 2 | 1 | 2 | 2 | **0** | 4 | 7 |
| 3 | 3 | 0 | **0** | 3 | 4 | 5 |
| 4 | 0 | **0** | 1 | 0 | 0 | 0 |
| 5 * | 0 | 3 | 0 | 7 | 6 | 4 |
| 6 + | 0 | 1 | 1 | 4 | 4 | 4 |

3. Then we can find the third column. In this column the chosen zero is in the third row. We put a + before the third row, and we add a * to the third column:

| | 5* | | 5* | | ↓ | ↓ |
|---|---|---|---|---|---|---|
| 1+ | **0** | 2 | 0 | 6 | 8 | 1 |
| 2 | 1 | 2 | 2 | **0** | 4 | 7 |
| 3+ | 3 | 0 | **0** | 3 | 4 | 5 |
| 4 | 0 | **0** | 1 | 0 | 0 | 0 |
| 5 * | 0 | 3 | 0 | 7 | 6 | 4 |
| 6 + | 0 | 1 | 1 | 4 | 4 | 4 |

4. Now we can find the + before the first row (or before the third row, or before the sixth row), let us choose e.g. row 1. In this row there are no more zeros, only in the already considered columns, so we  do not reach new column, but we change the + before row 1 to *.

| | 5* | | 5* | | ↓ | ↓ |
|---|---|---|---|---|---|---|
| 1* | **0** | 2 | 0 | 6 | 8 | 1 |
| 2 | 1 | 2 | 2 | **0** | 4 | 7 |
| 3+ | 3 | 0 | **0** | 3 | 4 | 5 |
| 4 | 0 | **0** | 1 | 0 | 0 | 0 |
| 5 * | 0 | 3 | 0 | 7 | 6 | 4 |
| 6 + | 0 | 1 | 1 | 4 | 4 | 4 |

5. Now we find the + e.g. before the third row. In this row there is one more zero (so that its column is not considered yet), column 2. We write “3”, i.e. the number of third row above the second column, and change the sign before row 3 from + to *.

| | 5* | 3 | 5* | | ↓ | ↓ |
|---|---|---|---|---|---|---|
| 1* | **0** | 2 | 0 | 6 | 8 | 1 |
| 2 | 1 | 2 | 2 | **0** | 4 | 7 |
| 3* | 3 | 0 | **0** | 3 | 4 | 5 |
| 4 | 0 | **0** | 1 | 0 | 0 | 0 |
| 5 * | 0 | 3 | 0 | 7 | 6 | 4 |
| 6 + | 0 | 1 | 1 | 4 | 4 | 4 |

6. We can choose now row 6. Here there is no more zero only in the first column, we already dealt with it, nothing other happens only change the + to * before row 6.

| | 5* | 3 | 5* | | ↓ | ↓ |
|---|---|---|---|---|---|---|
| 1* | **0** | 2 | 0 | 6 | 8 | 1 |
| 2 | 1 | 2 | 2 | **0** | 4 | 7 |
| 3* | 3 | 0 | **0** | 3 | 4 | 5 |
| 4 | 0 | **0** | 1 | 0 | 0 | 0 |
| 5* | 0 | 3 | 0 | 7 | 6 | 4 |
| 6* | 0 | 1 | 1 | 4 | 4 | 4 |

7. We find the number above column 2 without a *. We choose it, write a star here, and put a + before row 4 (as the chosen zero in column 2 is in row 4).

| | 5* | 3* | 5* | | ↓ | ↓ |
|---|---|---|---|---|---|---|
| 1* | **0** | 2 | 0 | 6 | 8 | 1 |
| 2 | 1 | 2 | 2 | **0** | 4 | 7 |
| 3* | 3 | 0 | **0** | 3 | 4 | 5 |
| 4+ | 0 | **0** | 1 | 0 | 0 | 0 |
| 5* | 0 | 3 | 0 | 7 | 6 | 4 |
| 6* | 0 | 1 | 1 | 4 | 4 | 4 |

8. We find the + before row 4. We find the zeros in this row in new columns, also in the desired columns, so we write there the number of row 4.

| | 5* | 3* | 5* | 4 | 4↓ | 4↓ |
|---|---|---|---|---|---|---|
| 1* | **0** | 2 | 0 | 6 | 8 | 1 |
| 2 | 1 | 2 | 2 | **0** | 4 | 7 |
| 3* | 3 | **0** | ~~**0**~~ | 3 | 4 | 5 |
| 4* | 0 | ~~**0**~~ | 1 | 0 | **0** | 0 |
| 5* | 0 | 3 | **0** | 7 | 6 | 4 |
| 6* | 0 | 1 | 1 | 4 | 4 | 4 |

What did happen? We found an augmenting path from some row that did not have a chosen zero to a column that did not have a chosen zero. (From a row we travel to a column, from a column to a row, and so on. We can go from a row to a column, if there is a zero (of any kind) in the cross. We can travel from a column to a row, if there is a CHOSEN zero in the cross!) Let us search this path! We can search it backward.

- We arrived (e.g.) to column 5. We arrived here from row 4. Let us choose the cross (denote it by bold red number).
- In this (i.e. fourth) row there is an “old” chosen zero. We drop it. It is in column 2. We arrived into this column from row 3. Let us choose this zero (red bold number).
- In this (i.e. third) row there is an “old” chosen zero. We drop it. It is in column 3. We arrived here from row 5. Let us choose this zero (red bold number).

We dropped 2 zeros, and we got 3 new zeroes instead. Together with the remained 2 old chosen zeros we do have 5 independent zeros instead of 4 independent zeros.

**Note**: This is the „same”, as finding augmenting path in a bipartite graph for constructing a maximum matching.

- Vertices in the left side: the rows
- vertices in the right side: the columns
- edges: where there is zero in the cross in the table
- two zeros are independent: the same as two edges are independent (no common vertex).
- chosen zeros in the table: a matching in the graph
- augmenting path in the graph: adjacent edges so that: it is not in the matching – it is in the matching - it is not - it is in - etc.

**So:** if we have an augmenting path, then we can enlarge the matching (the number of independent zeros)**.**

What happens if we do not have augmenting path? Like here (after having executed the Kőnig algorithm again):

| | 6* | | 1* | | | ↓ |
|---|---|---|---|---|---|---|
| 1* | **0** | 2 | 0 | 6 | 8 | 1 |
| 2 | 1 | 2 | 2 | **0** | 4 | 7 |
| 3 | 3 | **0** | 0 | 3 | 4 | 5 |
| 4 | 0 | 0 | 1 | 0 | **0** | 0 |
| 5* | 0 | 3 | **0** | 7 | 6 | 4 |
| 6* | 0 | 1 | 1 | 4 | 4 | 4 |

We can realize that there are three rows with * and there are only 2 columns with * .
It means that in the rows with * there are zeros only in the columns with *.

In other words:

- let us cover the rows with no star
- let us cover the columns with star

In this way we use only 5 covering lines to cover all zeros.
In the language of the bipartite graph: we cover all edges by exactly so many vertices like the number of the matching. (This proves the Kőnig- Egerváry theorem.)

**Note**:
1. During the Hungarian method, sometimes we need many transformations, sometimes few. We do not know in advance.
2. We can give easily an estimate for the sum of the numbers in the table, in each step. More exactly the sum reduces by n times epsilon times n-rho, if rho < n is the number of covering lines.
3. So the method is finite.
4. Total Step-number: Not more than $O(n^4)$.

**Some exercises for practice:**
1.

| | | | | | |
|---|---|---|---|---|---|
| 2 | 0 | 3 | 1 | 5 | 0 |
| 0 | 2 | 0 | 5 | 4 | 7 |
| 0 | 0 | 0 | 0 | 0 | 5 |
| 2 | 0 | 2 | 4 | 7 | 6 |
| 5 | 3 | 4 | 7 | 6 | 8 |
| 6 | 1 | 2 | 4 | 4 | 4 |

2.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 0 | 3 | 1 | 5 | 0 | 0 | 0 |
| 0 | 2 | 0 | 5 | 4 | 7 | 1 | 4 |
| 0 | 0 | 0 | 0 | 0 | 5 | 7 | 8 |
| 2 | 3 | 2 | 4 | 7 | 6 | 8 | 9 |
| 4 | 1 | 1 | 2 | 1 | 1 | 1 | 1 |
| 6 | 1 | 2 | 4 | 4 | 4 | 6 | 1 |
| 7 | 1 | 3 | 2 | 4 | 5 | 6 | 5 |
| 5 | 1 | 4 | 2 | 5 | 3 | 1 | 8 |

3. Let us start with the following system of independent zeros. Is it maximal? If not, augment it by the Kőnig algorithm. And, solve the assignment problem also for this exercise.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | **0** | 3 | 1 | 5 | 0 | 0 | 5 |
| **0** | 2 | 0 | 5 | 4 | 7 | 0 | 4 |
| 0 | 5 | **0** | 0 | 0 | 5 | 7 | 8 |
| 0 | 3 | 0 | 4 | 7 | **0** | 8 | 9 |
| 0 | 1 | 0 | 2 | 1 | 1 | 1 | **0** |
| 0 | 1 | 0 | 4 | 4 | 0 | 6 | 0 |
| 0 | 1 | 3 | **0** | 4 | 5 | 6 | 5 |
| 6 | 0 | 4 | 0 | 5 | 3 | 1 | 8 |