**Duy Tan University**

**Computer Science for Practicing Engineers**

**Travelling sale man problem**

**CMU SE 252 BIS**

**Team members:**

1. Lê Đình Nhật Khánh: 100%
2. Lữ Thanh Vĩnh: 100%
3. Trần Văn Thức: 100%
4. Huỳnh Bá Cường: 80%
5. Lý Bảo Khánh: 100%
6. Nguyễn Tùng Dương: 80%
7. **Example data:**

N = 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **@** | **A** | **B** | **C** | **D** | **E** |
| **A** | - | 7 | 6 | 8 | 4 |
| **B** | 7 | - | 8 | 5 | 6 |
| **C** | 6 | 8 | - | 9 | 7 |
| **D** | 8 | 5 | 9 | - | 8 |
| **E** | 4 | 6 | 7 | 8 | - |

Result:

* **Path: A 🡪 E 🡪 B 🡪 D 🡪 C 🡪 A**
* **Cost: 4 + 6 + 5 + 9 +6 = 30**

1. S**tep by step solving:**

**Step 1: Row minimization**

Row = row – minRow

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **@** | **A** | **B** | **C** | **D** | **E** |
| **A** | - | 3 | 2 | 4 | 0 |
| **B** | 2 | - | 3 | 0 | 1 |
| **C** | 0 | 2 | - | 3 | 1 |
| **D** | 3 | 0 | 4 | - | 3 |
| **E** | 0 | 2 | 3 | 4 | - |

**Step 2: Column minimization**

Column = Column – Column

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **@** | **A** | **B** | **C** | **D** | **E** |
| **A** | - | 3 | 0 | 4 | 0 |
| **B** | 2 | - | 1 | 0 | 1 |
| **C** | 0 | 2 | - | 3 | 1 |
| **D** | 3 | 0 | 2 | - | 3 |
| **E** | 0 | 2 | 1 | 4 | - |

**Step 3: Calculate penalties of all 0’s**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **@** | **A** | **B** | **C** | **D** | **E** |
| **A** | - | 3 | 0 (1) | 4 | 0 (1) |
| **B** | 2 | - | 1 | 0 (4) | 1 |
| **C** | 0 (1) | 2 | - | 3 | 1 |
| **D** | 3 | 0 (4) | 2 | - | 3 |
| **E** | 0 (1) | 2 | 1 | 4 | - |

**Step 4: Delete the row + column that contains the point with the highest penalty ([2][4])**

**We have B 🡪 D**

**Reduced matrix:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **@** | **A** | **B** | **C** | **E** |
| **A** | - | 3 | 0 | 0 |
| **C** | 0 | 2 | - | 1 |
| **D** | 3 | -  (Travelled) | 2 | 3 |
| **E** | 0 | 2 | 1 | - |

**Step 5: Minimize row and column**

**(To make sure that each row and column contains at least one 0)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **@** | **A** | **B** | **C** | **E** |
| **A** | - | 1 | 0 | 0 |
| **C** | 0 | 0 | - | 1 |
| **D** | 1 | - | 0 | 1 |
| **E** | 0 | 0 | 1 | - |

**Step 6: Calculate penalties**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **@** | **A** | **B** | **C** | **E** |
| **A** | - | 1 | 0 (0) | 0 (1) |
| **C** | 0 (0) | 0 (0) | - | 1 |
| **D** | 1 | - | 0 (1) | 1 |
| **E** | 0 (0) | 0 (0) | 1 | - |

**Max penalty value is 1**

**Delete [1][4]**

**We have: A 🡪 E**

**Reduced matrix:**

|  |  |  |  |
| --- | --- | --- | --- |
| **@** | **A** | **B** | **C** |
| **C** | 0 | 0 | - |
| **D** | 1 | - | 0 |
| **E** | -  (Travelled) | 0 | 1 |

**Step 7: Calculate penalties**

|  |  |  |  |
| --- | --- | --- | --- |
| **@** | **A** | **B** | **C** |
| **C** | 0 (1) | 0 (0) | - |
| **D** | 1 | - | 0 (2) |
| **E** | -  (Travelled) | 0 (1) | 1 |

**Max penalty value is 2**

**Delete [2][3]**

**We have: D 🡪 C**

**Reduced matrix:**

|  |  |  |
| --- | --- | --- |
| **@** | **A** | **B** |
| **C** | 0 (0) | 0 (0) |
| **E** | - | 0 (0) |

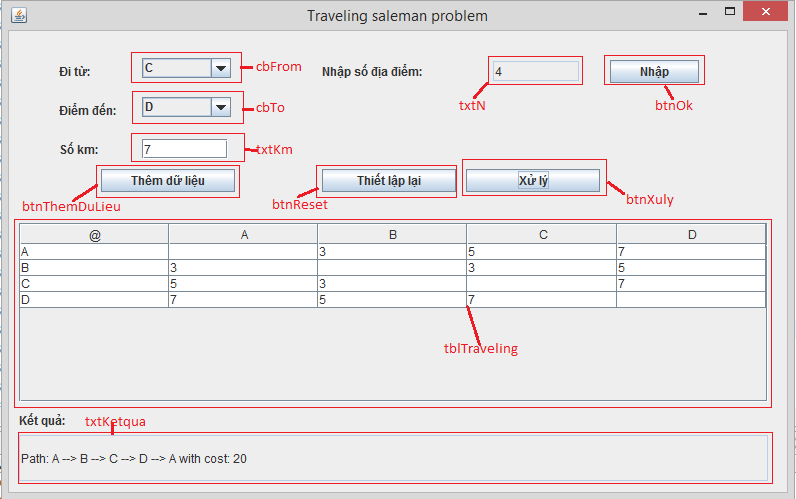
**We can't travel A 🡪 E again so it’s must be C 🡪 A and E 🡪 B**

**So now we have: A🡪 E, C🡪 A, E🡪 B, D 🡪 C, B 🡪 D**

**Sort: A 🡪 E 🡪 B 🡪 D 🡪 C 🡪 A**

**Cost: 30**

III) How to use the Traveling sale man problem GUI:



* Enter the number of places you want to travel at text field txtN
* Click button btnOk to confirm your choice
* The system will automatically create a table (tblTraveling) that contain all the places
* Choose the place at combo box cbFrom and cbTo to set the distance from cbFrom to cbTo
* Enter the distance from cbFrom to cbTo at text field txtKM
* Click the button btnThemDuLieu to confirm your choice (the data will be show at the table tblTraveling below)
* Click button btnXuly to solve your problem and the result will be displayed at text field txtKetqua