Python installing alternatives

Python:

https://www.python.org/downloads/

Pycharm IDE:

https://www.jetbrains.com/pycharm/

Anaconda:

https://www.anaconda.com/distribution/#download-section

Google Colab:

https://colab.research.google.com/

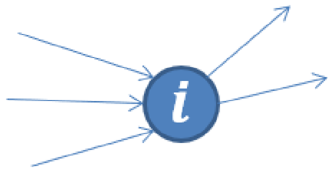
Libraries:

- Numpy
- Scipy
- Matplotlib

Basic utils

#EX0: Understanding model needs

$$\min \sum_{ij \in A} c_{ij} x_{ij}$$



$$\sum_{\substack{j \\ ij \in A}} x_{ij} - \sum_{\substack{j \\ ji \in A}} x_{ji} = b_i \quad \forall i \in N$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall ij \in A$$

C: cost vector, **dimension: arcs**
b: linear equality constraint, **dimension: nodes**
lb: lower bound, dimension: **arcs**
ub: upper ound, dimension: **arcs**

NA: node arc matrix, **dimension: nodes x arcs**

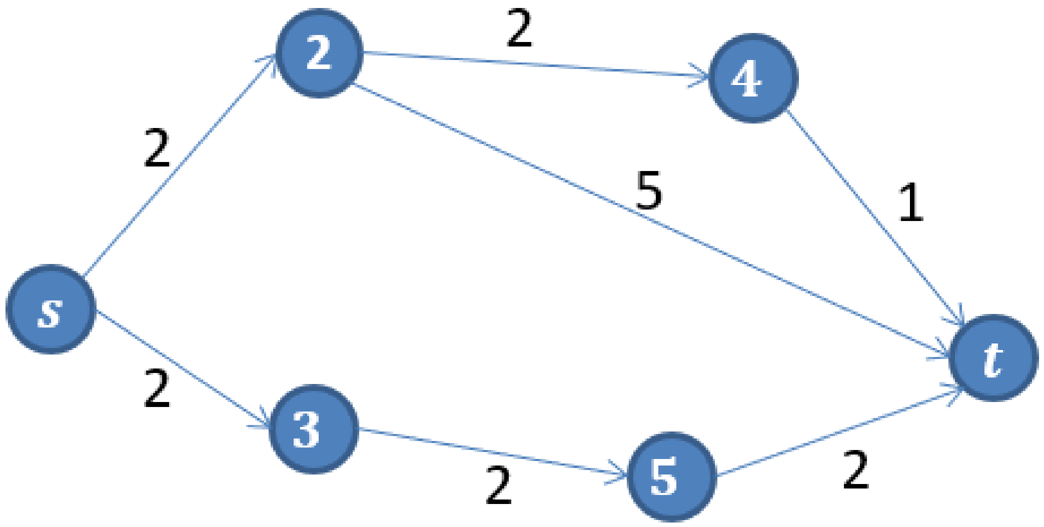
#EX0: Build a function that transforms Node-Node matrix (incidence matrix) to Node-Arc matrix

* Need help? get the solution from repository: https://github.com/rmaranzana/oss11_basic_examples

Shortest Path

#EX1: Basic example SP

1) Understand the MCF model formulation.
2) Try to translate the math to the python code:
    1) Numpy library for matrix operations
    2) Scipy library for LP optimization
3) Learn to use scipy.linprog
4) Understand the outputs
5) What's the shortest path?

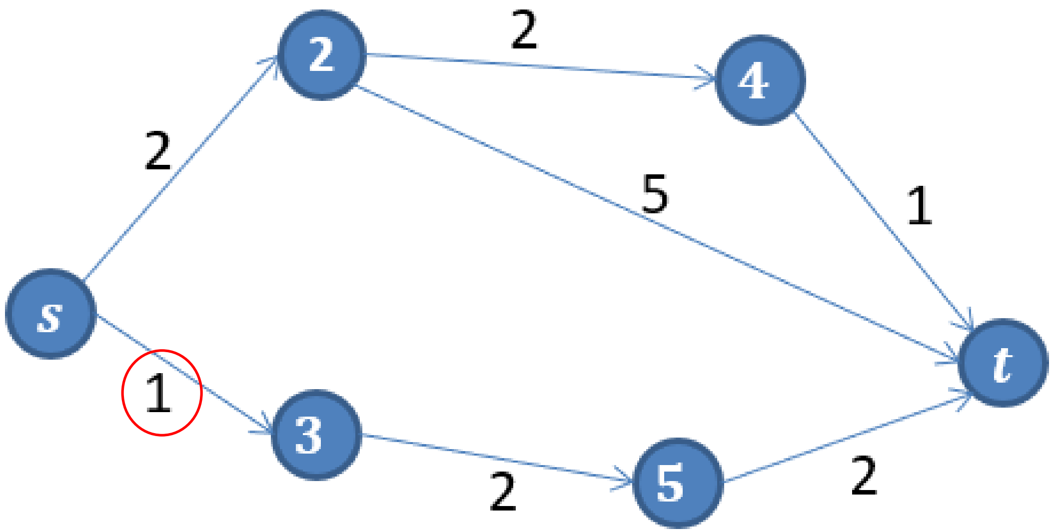


* Need help? get the solution from repository: https://github.com/rmaranzana/oss11_basic_examples

Shortest Path

#EX2: Basic example SP

1) Try the same procedure with this example
2) Check the solution
3) Try changing the method to SIMPLEX
4) Get the shortest path

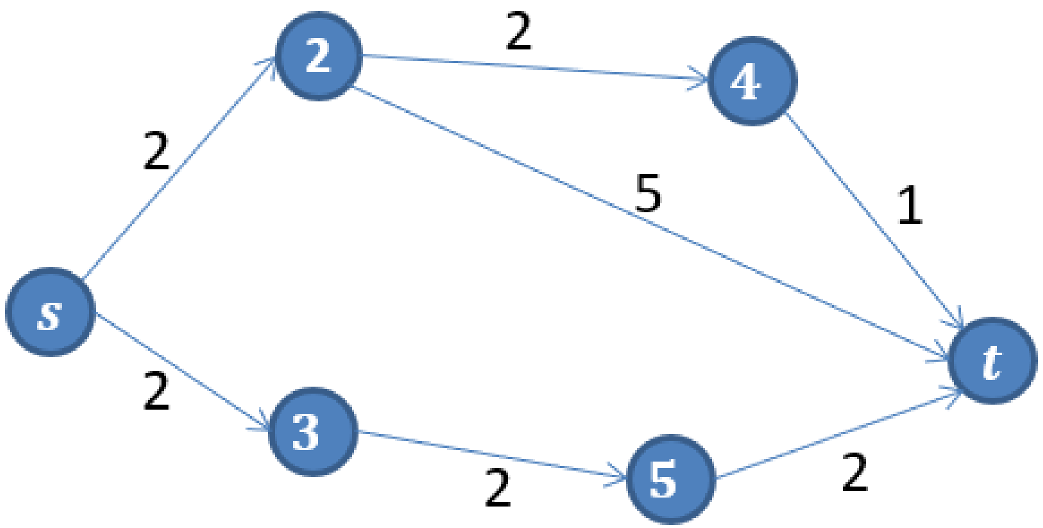


* Need help? get the solution from repository: https://github.com/rmaranzana/oss11_basic_examples

Shortest Path

#EX3 Basic example SP with Dijkstra algorithm

1) Implement the Dijkstra algorithm and check the #EX1 solution



Dijkstra algorithm

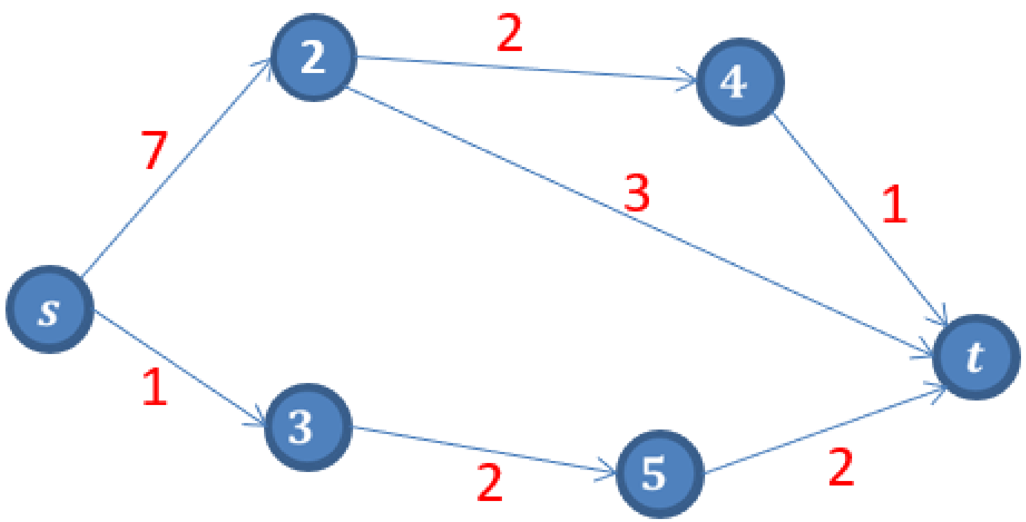
Pseudocode:

```
# Initialize the weights:
for each node_i in graph:
    weight(node_i) = inf
    prec[node_i] = 0
weight(initial_node) = 0

# Iterate:
while unexplored_nodes not empty:
    head = node with minimum weight in unexplored_nodes
    pop head node from unexplored_nodes

    for each neighbor_i of head:
        potential_weight = weight[head] + dist[head, neighbor_i]
        if potential_weight < weight[neighbor]:
            weight[neighbor] = potential weight
            prec[neighbor] = head
```

Maximum Flow

#EX3: Basic example SP

1) Understand the MCF model formulation.
2) Translate the math to python code. *(Pay attention to model boundaries, LB / UB)*
3) Understand the outputs
4) What's the maximum flow?
5) What's the minimum cut?

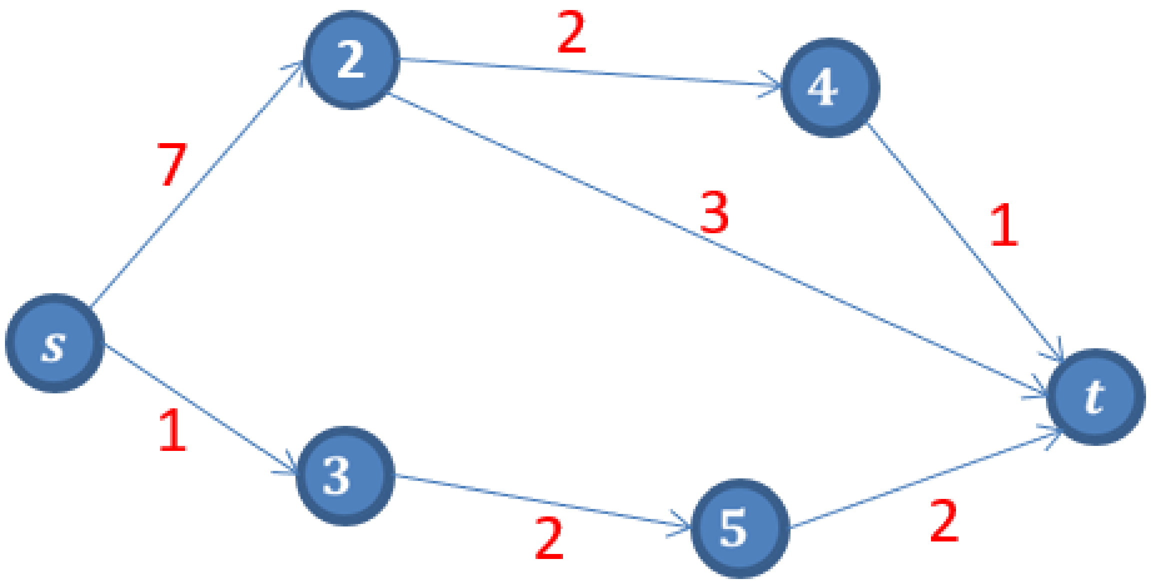


* Need help? get the solution from repository: https://github.com/rmaranzana/oss11_basic_examples

Maximum Flow

#EX4: Basic example SP with Ford-Fulkerson algorithm

1) Implement the Ford-Fulkerson algorithm and check the #EX3 solution



* Need help? get the solution from repository: https://github.com/rmaranzana/oss11_basic_examples

Ford-Fulkerson algorithm

Pseudocode:

```
residual_G = G for all edges(u, v) and 0 for all edges (v, u)

path = perform DFS(source, sink) to find an augmenting path

while path exists:
    for each edge in path:
            residual_G(u, v) = flow(u, v) - path max_capacity
            residual_G(v, u) = flow(v, u) + path max_capacity

    path = perform DFS(source, sink)
```

Transport

#EX5: Transport model example

1) Understand the MCF model formulation
2) Get the optimum number of units to be carried from factories to warehouses for each arc.