Student Name:
Student ID:

LabReSiD25
Hands-On 17

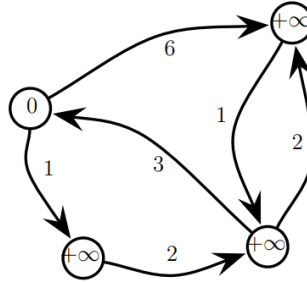Università
degli Studi di
Messina

Bellman-Ford: Distributed Shortest Path

Implementare in C, usando le fifo, l'algoritmo di Bellman-Ford per il grafo in figura.

`Synchronous Network with Weights:` $\mathcal{S} = (\{1, \dots, n\}, E_{\mathrm{cmm}}, A)$

`Distributed Algorithm:` DISTRIBUTED BELLMAN-FORD

`Alphabet:` $\mathbb{A} = \mathbb{R}_{>0} \cup \mathtt{null} \cup \{+\infty\}$

`Processor State:` $w = (\mathtt{parent}, \mathtt{dist})$, where

$\mathtt{parent} \in \{1, \dots, n\},$	initially: $\mathtt{parent}^{[j]} = j$ for all j
$\mathtt{dist} \quad \in \mathbb{A},$	initially: $\mathtt{data}^{[1]} = 0,$
	$\mathtt{data}^{[j]} = +\infty$ for all $j \neq 1$

`function` $\mathrm{msg}(w, i)$

  1: **if** round $< n$ **then**
  2:    **return** dist
  3: **else**
  4:    **return** null

`function` $\mathrm{stf}(w, y)$

  1: $i :=$ processor UID
  2: $k := \mathrm{arginf}\{y_j + a_{ji} \mid \text{for all } y_j \neq \mathtt{null}\}$
  3: **if** $(\mathtt{dist} < k)$ **then**
  4:    **return** $(\mathtt{parent}, \mathtt{dist})$
  5: **else**
  6:    **return** $(k, y_k + a_{ki})$