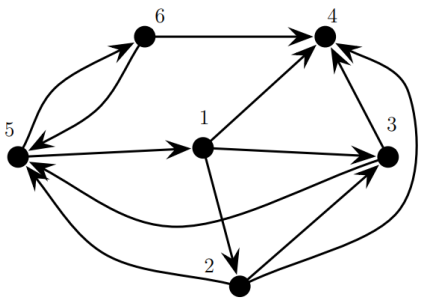
Student Name:
Student ID:

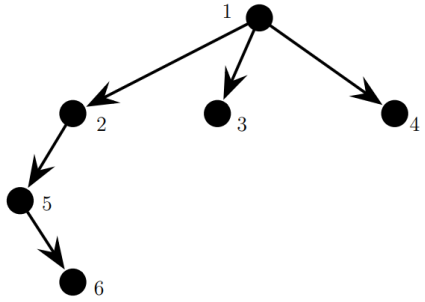
LabReSiD25
Hands-On 16

Università degli Studi di Messina

Leader Election con processi e fifo

Implementare Floodmax per il grafo in figura (a sinistra) ed LCR per un grafo ciclico di ordine 5.





`Alphabet`: $\mathbb{A} = \{1, \ldots, n\} \cup \{\texttt{null}\}$

`Processor State`: $w = (\texttt{my-id}, \texttt{max-id}, \texttt{leader}, \texttt{round})$, where

`my-id`	$\in \{1, \ldots, n\}$,	initially: $\texttt{my-id}^{[i]} = i$ for all i
`max-id`	$\in \{1, \ldots, n\}$,	initially: $\texttt{max-id}^{[i]} = i$ for all i
`leader`	$\in \{\texttt{false}, \texttt{true}, \texttt{unknwn}\}$,	initially: $\texttt{leader}^{[i]} = \texttt{unknwn}$ for all i
`round`	$\in \{0, 1, \ldots, \text{diam}(\mathcal{S})\}$,	initially: $\texttt{round}^{[i]} = 0$ for all i

`function` msg(w, i)

1: **if** `round` $<$ diam$(\mathcal{S})$ **then**
2: **return** `max-id`
3: **else**
4: **return** `null`

`function` stf(w, y)

1: `new-id`:= max{`max-id`, largest identifier in y}
2: **case**
3: `round` $<$ diam$(\mathcal{S})$: `new-lead` := `unknwn`
4: `round` $=$ diam$(\mathcal{S})$ AND `max-id` $=$ `my-id`: `new-lead` := `true`
5: `round` $=$ diam$(\mathcal{S})$ AND `max-id` $>$ `my-id`: `new-lead` := `false`
6: **return** (`my-id`, `new-id`, `new-lead`, `round` $+1$)

`Synchronous Network:` ring digraph

`Distributed Algorithm:` LCR

`Alphabet:` $\mathbb{A} = \{1, \ldots, n\} \cup \{\texttt{null}\}$

`Processor State:` $w = (\texttt{my-id}, \texttt{max-id}, \texttt{leader}, \texttt{snd-flag})$, where

`my-id`	$\in \{1, \ldots, n\}$,	initially: $\texttt{my-id}^{[i]} = i$ for all i
`max-id`	$\in \{1, \ldots, n\}$,	initially: $\texttt{max-id}^{[i]} = i$ for all i
`leader`	$\in \{\texttt{true}, \texttt{false}, \texttt{unknwn}\}$,	initially: $\texttt{leader}^{[i]} = \texttt{unknwn}$ for all i
`snd-flag`	$\in \{\texttt{true}, \texttt{false}\}$,	initially: $\texttt{snd-flag}^{[i]} = \texttt{true}$ for all i

`function` msg(w, i)

```
1: if snd-flag = true then
2:    return max-id
3: else
4:    return null
```

`function` stf(w, y)

```
1: case
2:   (y contains only null msgs) OR (largest identifier in y < my-id):
3:       new-id := max-id
4:       new-lead := leader
5:       new-snd-flag := false
6:   (largest identifier in y = my-id):
7:       new-id := max-id
8:       new-lead := true
9:       new-snd-flag := false
10:  (largest identifier in y > my-id):
11:      new-id := largest identifier in y
12:      new-lead := false
13:      new-snd-flag := true
14: return (my-id, new-id, new-lead, new-snd-flag)
```
