Distributed and Parallel Computing

Lecture 13

Alan P. Sexton

University of Birmingham

Spring 2019

The Echo Algorithm

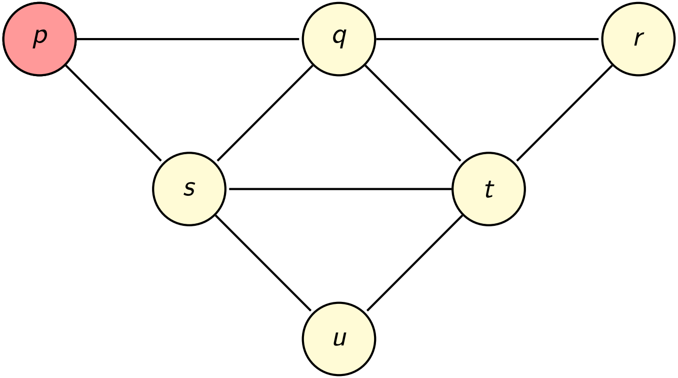
A wave, but not a traversal algorithm (so no *tokens* involved), *Echo* is a centralized algorithm (i.e. one initiator only) for undirected networks.

- Initiator sends message to all neighbours
- When a non-initiator *first* receives a message
    - It makes the sender its parent
    - It sends a message to all neighbours except its parent
- When a non-initiator has received messages from *all* its neigbours
    - It sends a message to its parent
- When the initiator has received messages from all its neighbours, it *decides* and the algorithm terminates

This algorithm builds a spanning tree

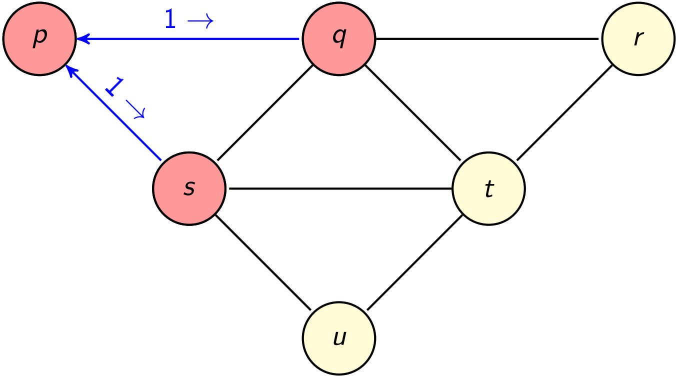
Example Execution of the Echo Algorithm

Here if messages are sent out immediately after each other without waiting to receive a message in between sends, they are given the same number. In reality, real messages are not sent simultaneously.



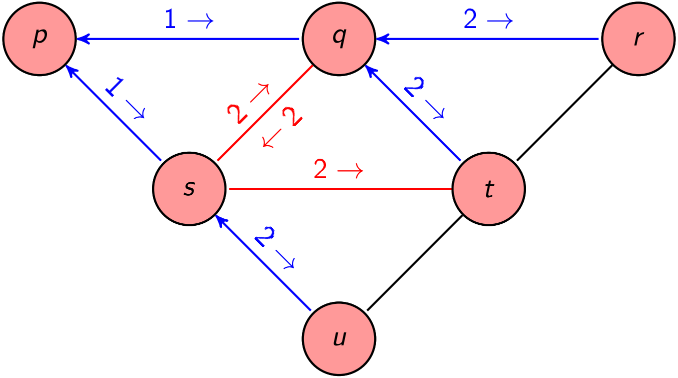
Example Execution of the Echo Algorithm

Here if messages are sent out immediately after each other without waiting to receive a message in between sends, they are given the same number. In reality, real messages are not sent simultaneously.



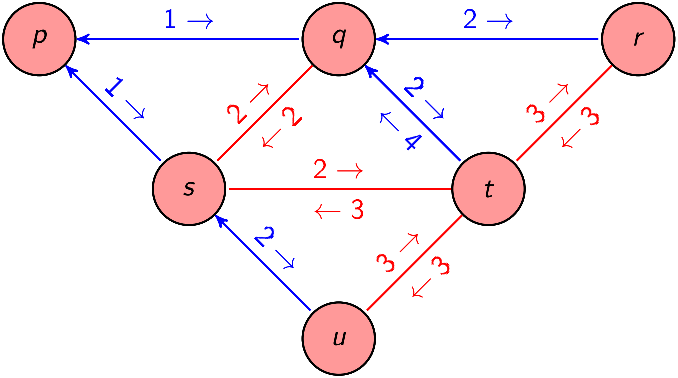
Example Execution of the Echo Algorithm

Here if messages are sent out immediately after each other without waiting to receive a message in between sends, they are given the same number. In reality, real messages are not sent simultaneously.



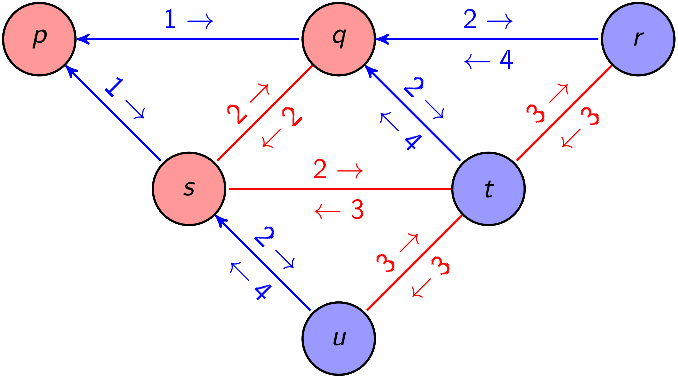
Example Execution of the Echo Algorithm

Here if messages are sent out immediately after each other without waiting to receive a message in between sends, they are given the same number. In reality, real messages are not sent simultaneously.



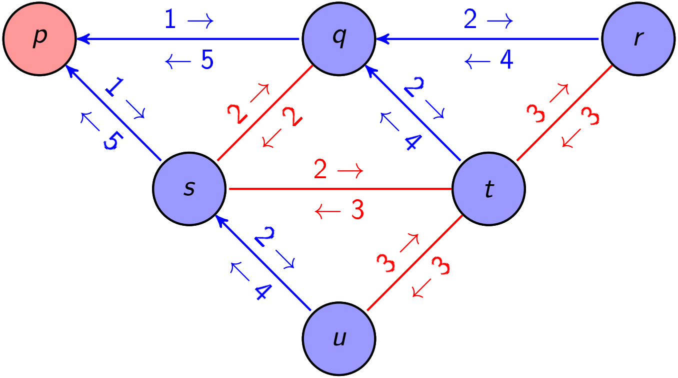
Example Execution of the Echo Algorithm

Here if messages are sent out immediately after each other without waiting to receive a message in between sends, they are given the same number. In reality, real messages are not sent simultaneously.



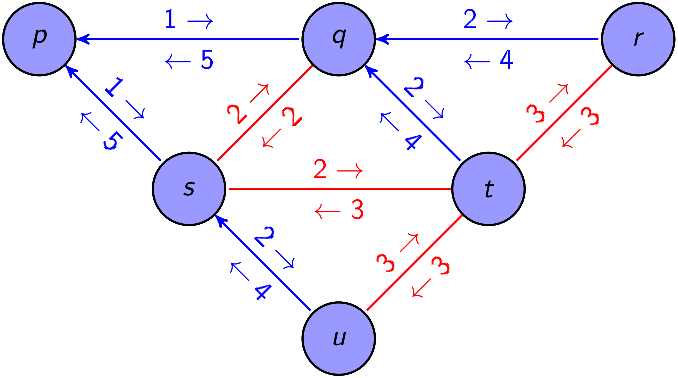
Example Execution of the Echo Algorithm

Here if messages are sent out immediately after each other without waiting to receive a message in between sends, they are given the same number. In reality, real messages are not sent simultaneously.


p
q
r
s
t
u
1 →
← 5
2 →
← 4
1 ↘
↖ 5
2 ↗
↙ 2
2 ↘
↖ 4
3 ↗
↙ 3
2 →
← 3
2 ↘
↖ 4
3 ↗
↙ 3

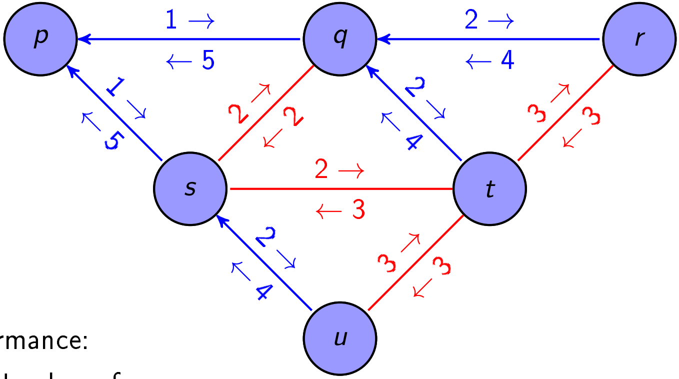
Example Execution of the Echo Algorithm

Here if messages are sent out immediately after each other without waiting to receive a message in between sends, they are given the same number. In reality, real messages are not sent simultaneously.


p
q
r
s
t
u
1 →
← 5
2 →
← 4
1 ↘
↖ 5
2 ↗
↙ 2
2 ↘
↖ 4
3 ↗
↙ 3
2 →
← 3
2 ↘
↖ 4
3 ↗
↙ 3

Example Execution of the Echo Algorithm

Here if messages are sent out immediately after each other without waiting to receive a message in between sends, they are given the same number. In reality, real messages are not sent simultaneously.

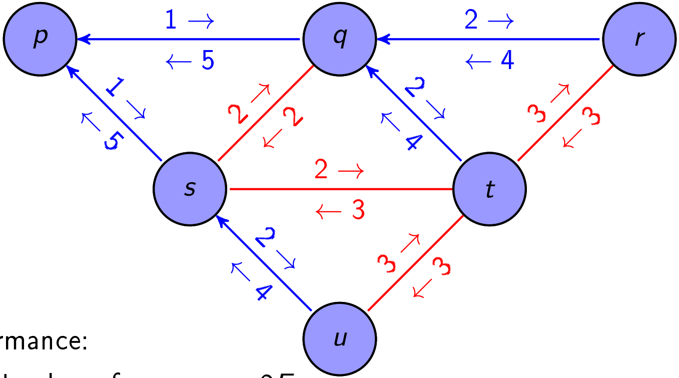


Performance:

- Number of messages:
- Worst case time to complete:

Example Execution of the Echo Algorithm

Here if messages are sent out immediately after each other without waiting to receive a message in between sends, they are given the same number. In reality, real messages are not sent simultaneously.

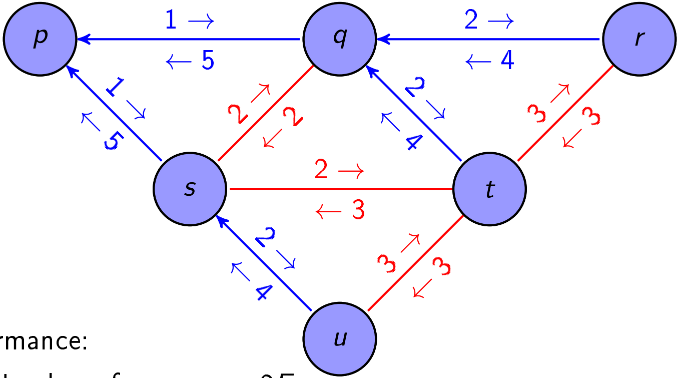


Performance:

- Number of messages: $2E$
- Worst case time to complete:

Example Execution of the Echo Algorithm

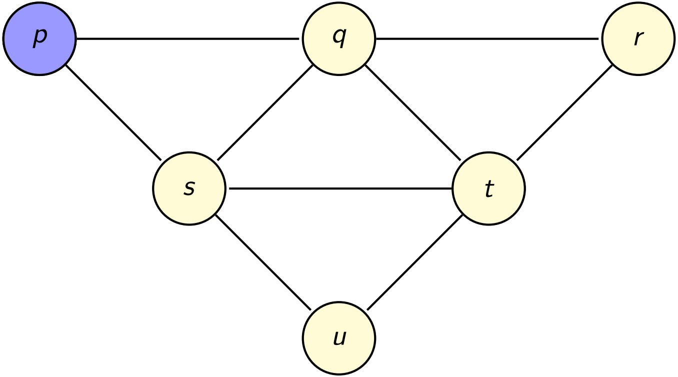
Here if messages are sent out immediately after each other without waiting to receive a message in between sends, they are given the same number. In reality, real messages are not sent simultaneously.



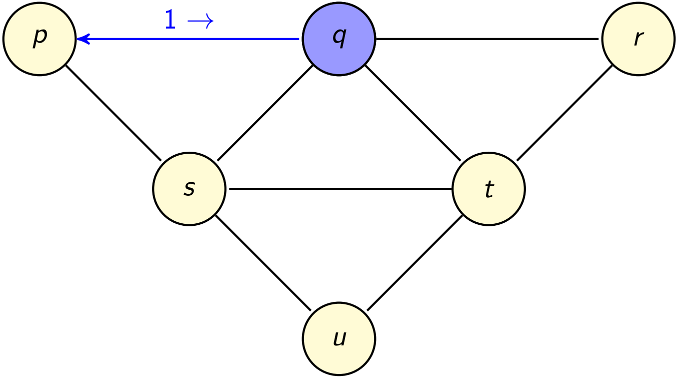
Performance:

- Number of messages: $2E$
- Worst case time to complete: $2N - 2$ time units

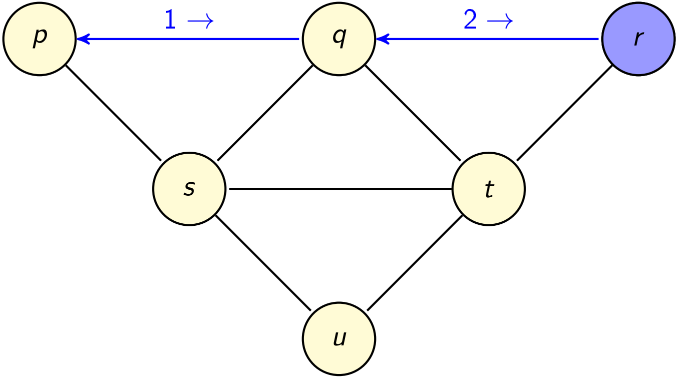
Tarry Algorithm considered as an Echo Algorithm



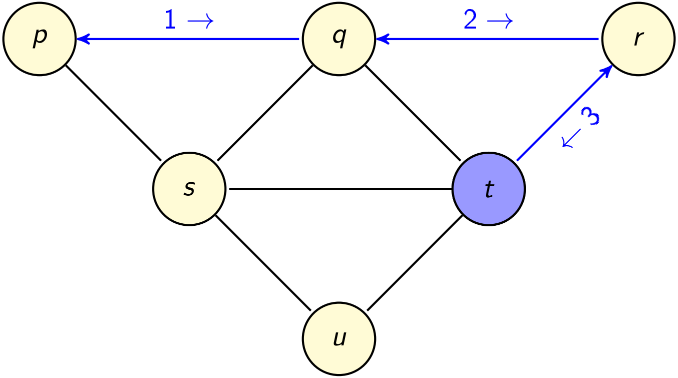
Tarry Algorithm considered as an Echo Algorithm



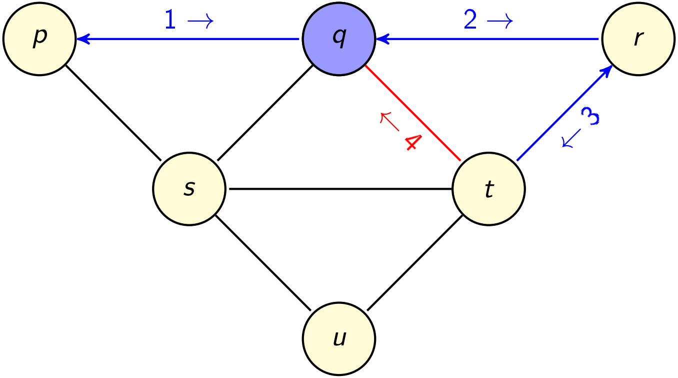
Tarry Algorithm considered as an Echo Algorithm



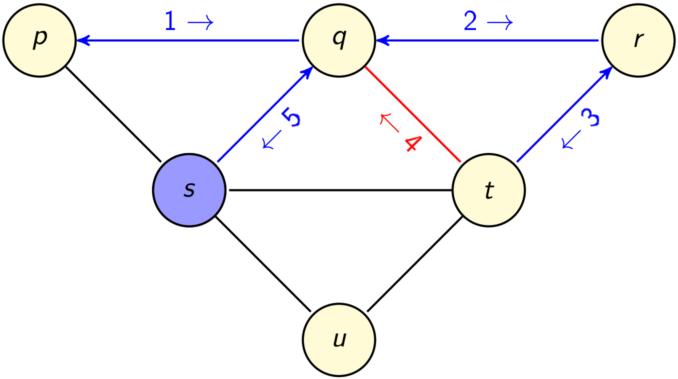
Tarry Algorithm considered as an Echo Algorithm



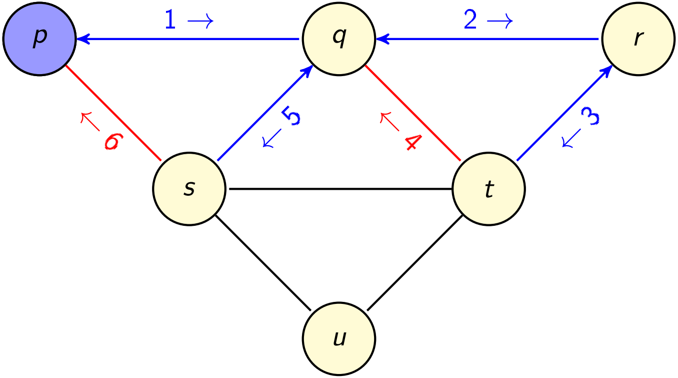
Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
2 →
3 ↙
↖ 4

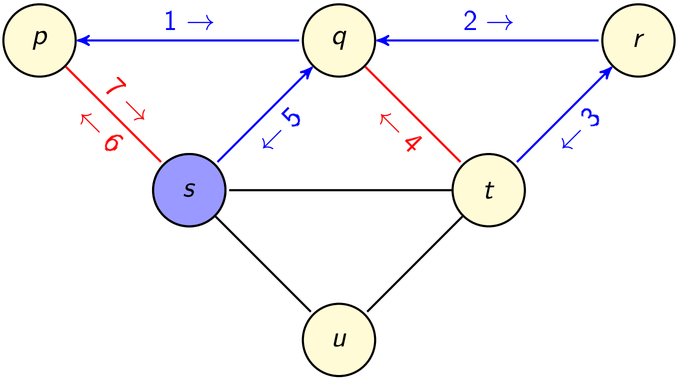
Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
2 →
↙ 3
↖ 4
↙ 5

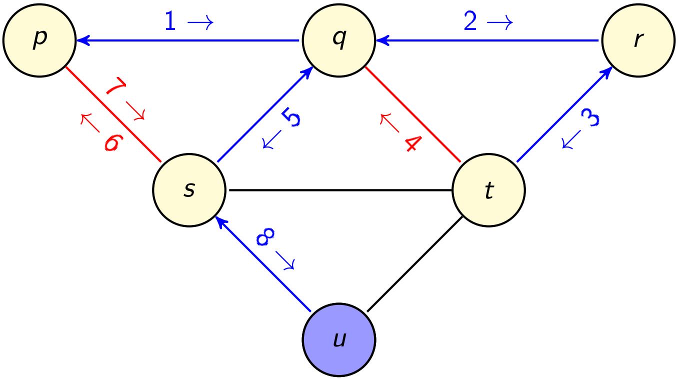
Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
2 →
↙ 3
↖ 4
↙ 5
↖ 6

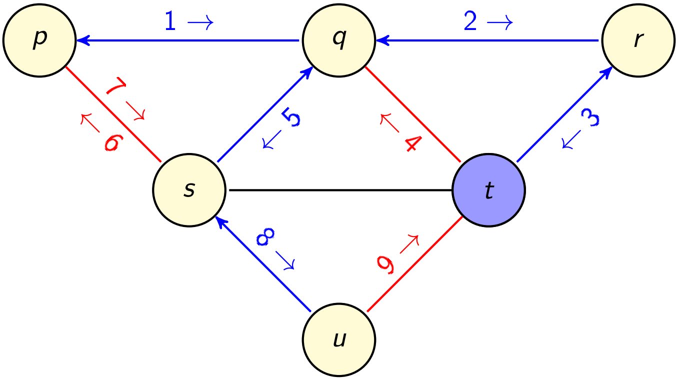
Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
2 →
3 ↙
4 ↖
5 ↙
6 ↖
7 ↘

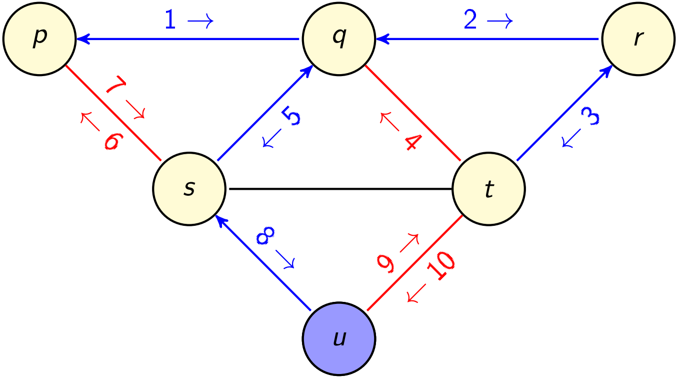
Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
2 →
7 →
← 6
5 ↙
← 4
3 ↙
8 ↘

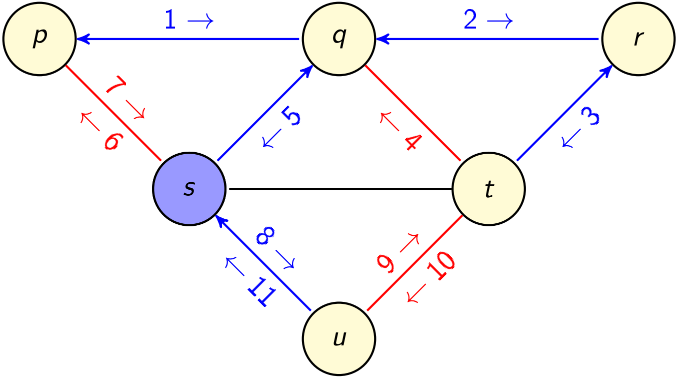
Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
2 →
3 ↙
4 ↖
5 ↙
6 ↖
7 ↘
8 ↘
9 ↗

Tarry Algorithm considered as an Echo Algorithm

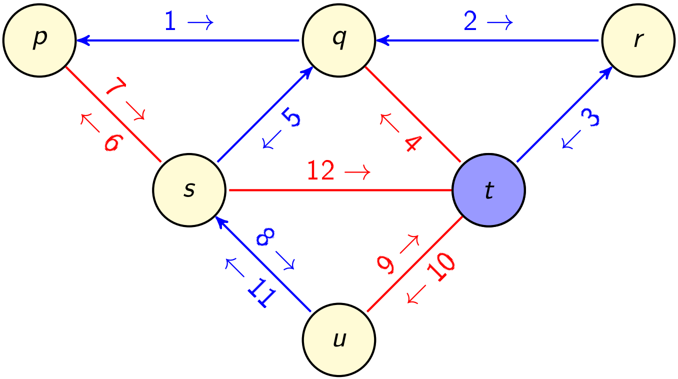

p
q
r
s
t
u
1 →
2 →
← 3
← 4
← 5
← 6
7 →
8 →
9 →
← 10

Tarry Algorithm considered as an Echo Algorithm



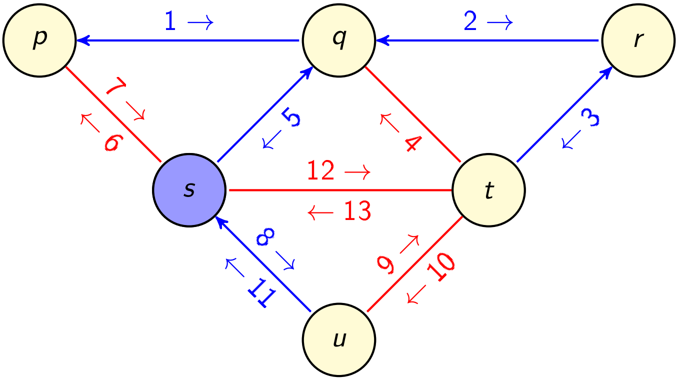
p
q
r
s
t
u
1 →
2 →
7 →
← 6
5
↙
← 4
3
↙
8 →
← 11
9 →
10
↙

Tarry Algorithm considered as an Echo Algorithm

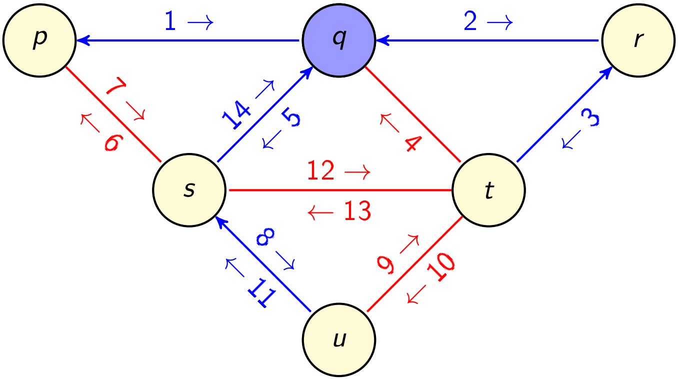


p
q
r
s
t
u
1 →
2 →
7 →
← 6
5
↙
← 4
↙
3
12 →
8 →
← 11
9 →
10
↙

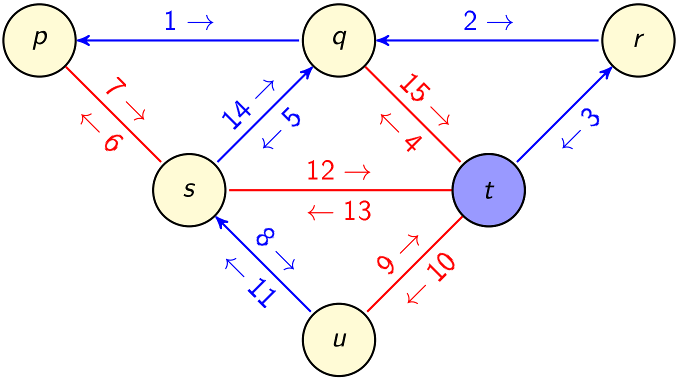
Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
2 →
7 →
← 6
↙ 5
↖ 4
↙ 3
12 →
← 13
8 →
← 11
9 →
↙ 10

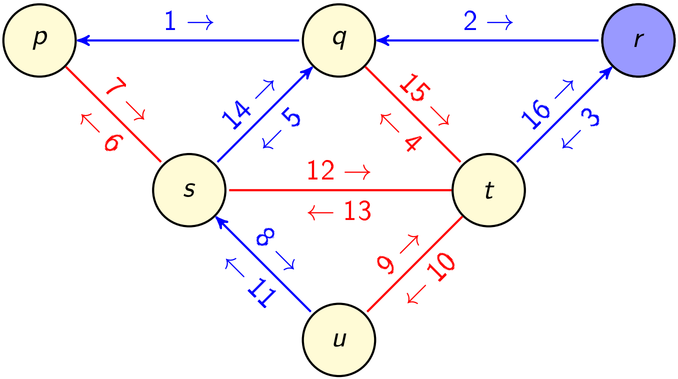
Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
2 →
3
4
5
6
7 →
8 →
9 →
10
← 11
12 →
← 13
14 →

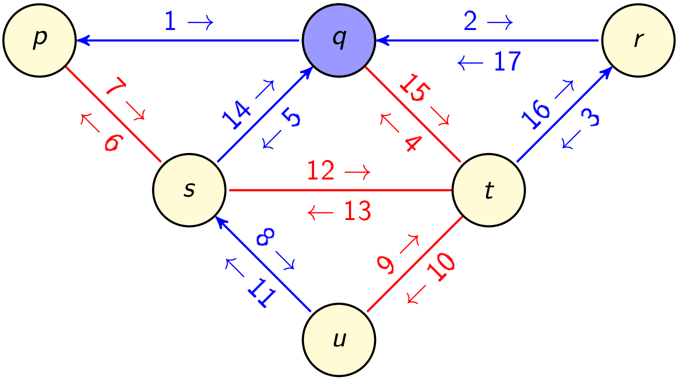
Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
2 →
7 →
← 6
14 →
← 5
15 →
← 4
← 3
12 →
← 13
8 →
← 11
9 →
← 10

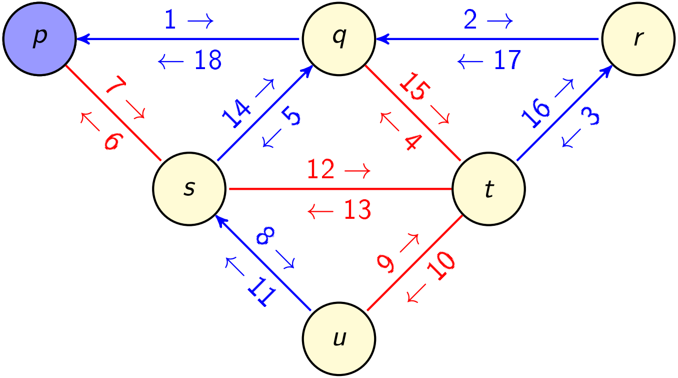
Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
2 →
7 →
← 6
14 →
← 5
15 →
← 4
16 →
← 3
12 →
← 13
8 →
← 11
9 →
← 10

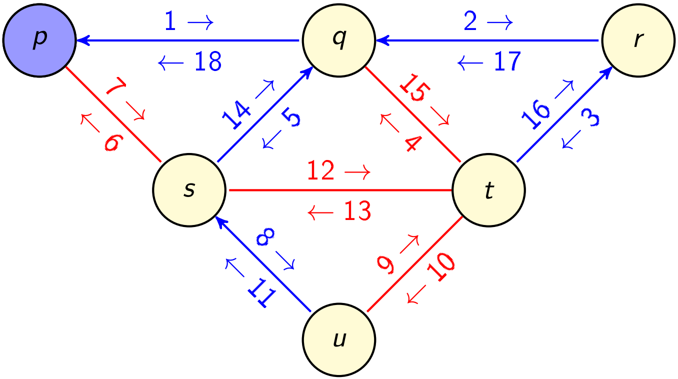
Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
2 →
← 17
7 →
← 6
14 →
← 5
15 →
← 4
16 →
← 3
12 →
← 13
8 →
← 11
9 →
← 10

Tarry Algorithm considered as an Echo Algorithm

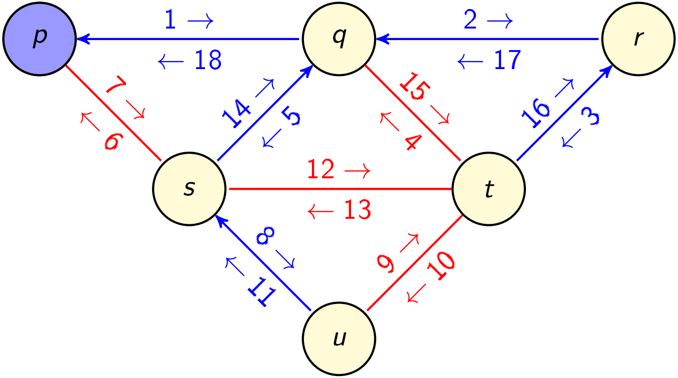

p
q
r
s
t
u
1 →
← 18
2 →
← 17
7 →
← 6
14 →
← 5
15 →
← 4
16 →
← 3
12 →
← 13
8 →
← 11
9 →
← 10

Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
← 18
2 →
← 17
7 →
← 6
14 →
← 5
15 →
← 4
16 →
← 3
12 →
← 13
8 →
← 11
9 →
← 10

- Every message trace of the execution of a Tarry algorithm is a possible message trace of the execution of an Echo algorithm

Tarry Algorithm considered as an Echo Algorithm


p
q
r
s
t
u
1 →
← 18
2 →
← 17
7 →
← 6
14 →
← 5
15 →
← 4
16 →
← 3
12 →
← 13
8 →
← 11
9 →
← 10

- Every message trace of the execution of a Tarry algorithm is a possible message trace of the execution of an Echo algorithm
- Exercise: Find a message trace of an Echo algorithm that is not the message trace of a Tarry algorithm