

```
void fastest_sort(vector<int>& A)
{
    std::vector<int> B;
    vector<int>::iterator half;
    int item = 0;

    //Base case
    B.push_back(A.back());
    A.pop_back();

    // Other cases
    while(!A.empty())
    {
        item = A.back();

        half = find_insert_pos(B,item);

        if (*half < item)
            B.insert (half+1,item);
        else
            B.insert (half,item);

        A.pop_back();
    }
    A=B;
}
```

```
vector<int>::iterator find_insert_pos(vector<int>& B,int valor)
{
    vector<int>::iterator inicio = B.begin();
    vector<int>::iterator fin      = B.end()-1;
    vector<int>::iterator mitad;

    while (inicio <= fin)
    {
        mitad = inicio + (fin-inicio)/2;

        if (*mitad == valor) // Si es numero repetido
            break;

        if (*mitad < valor) // Recorrer inicio al centro
            inicio = mitad + 1;

        else // recorrer final hacia la izquierda
            fin      = mitad - 1;
    }
    return mitad;
}
```



7

2

1

-4

0

9

13

21

Inicio

Final

Na



7

2

1

-4

0

9

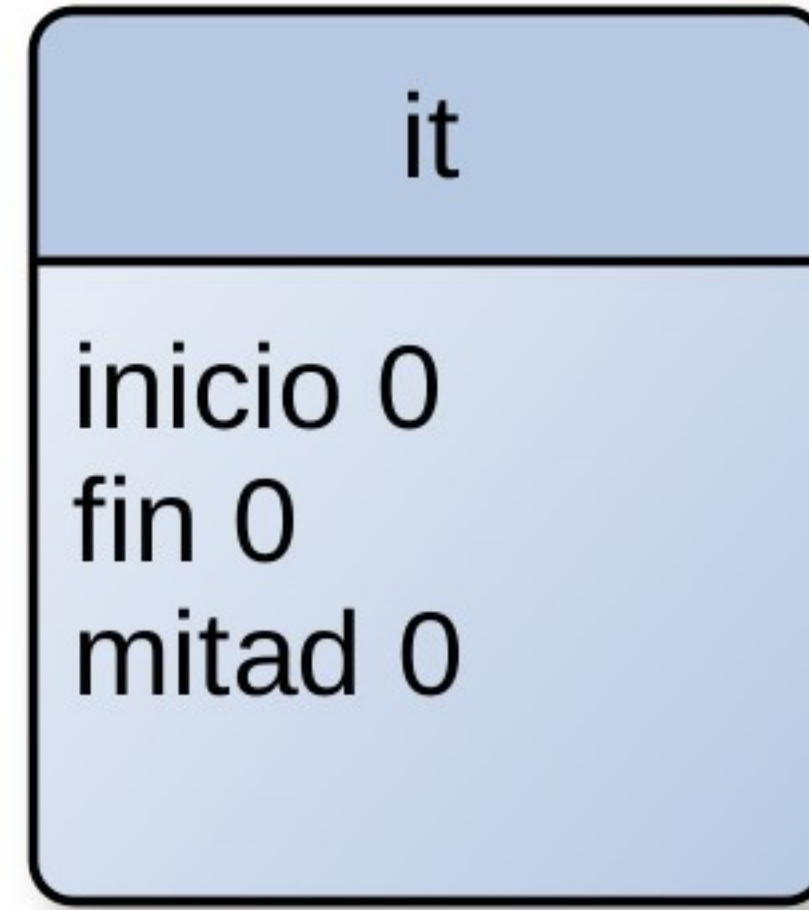
13

21

Inicio

Final

Na



7

2

1

-4

0

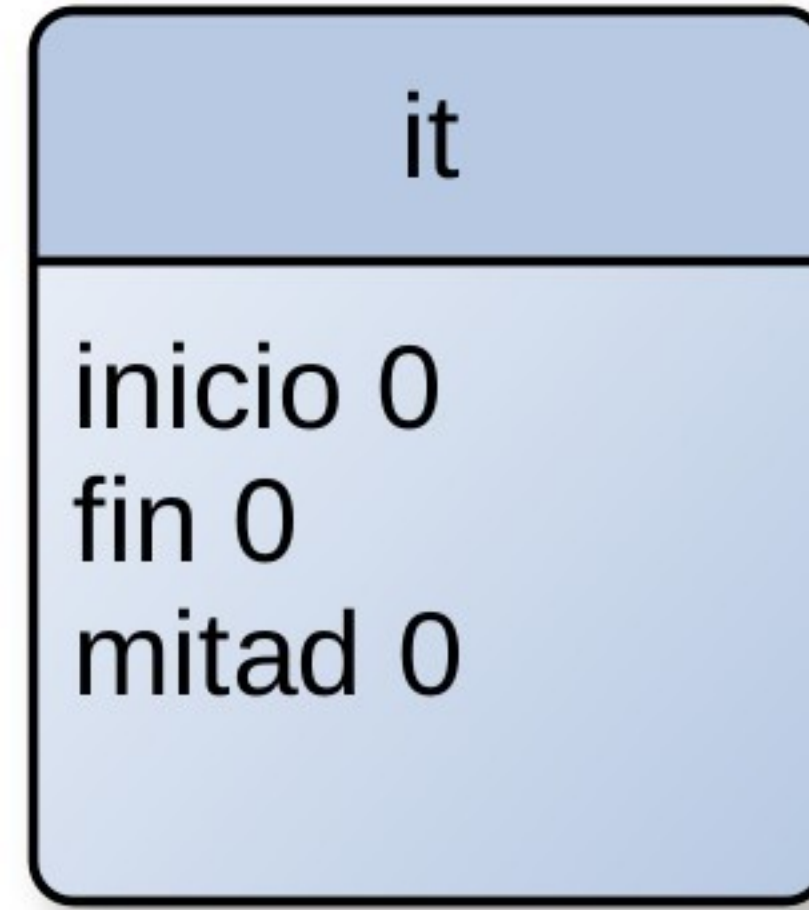
9

13

Inicio

Final

21



7

2

1

-4

0

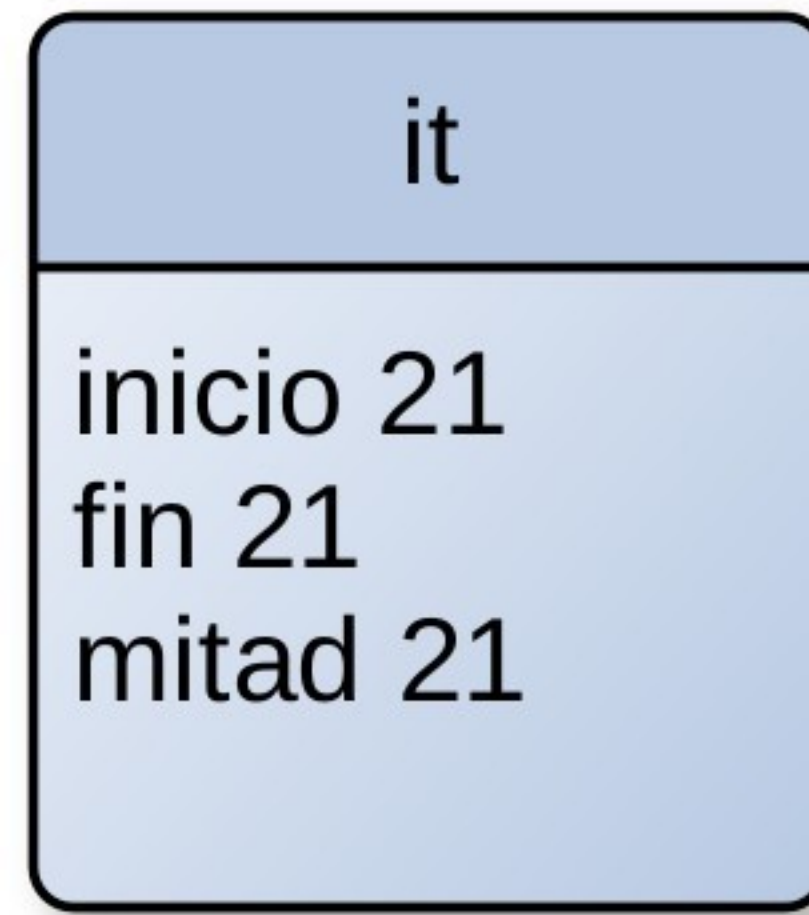
9

13

Inicio

Final

21



7

2

1

-4

0

9

13

Inicio

Mitad

Final

21



7

2

1

-4

0

9

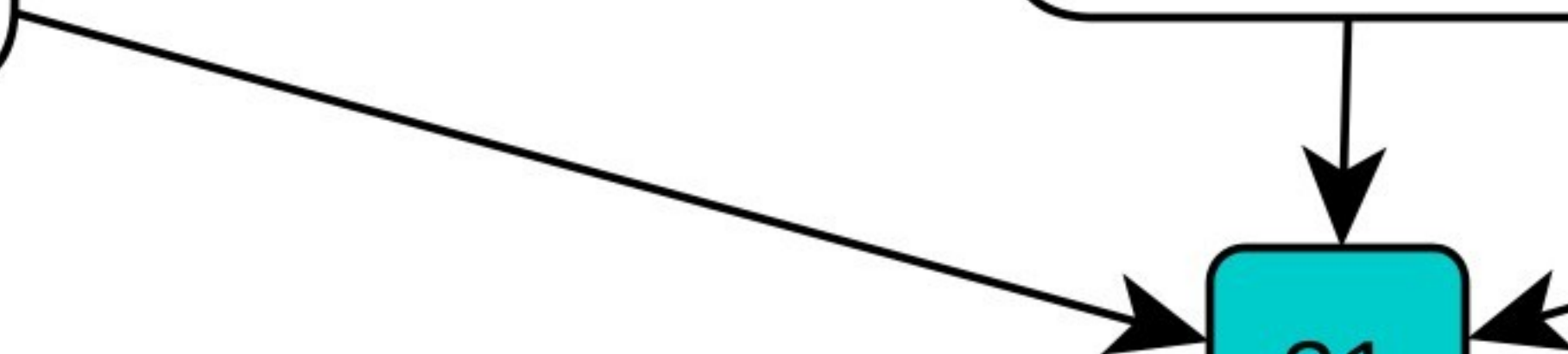
13

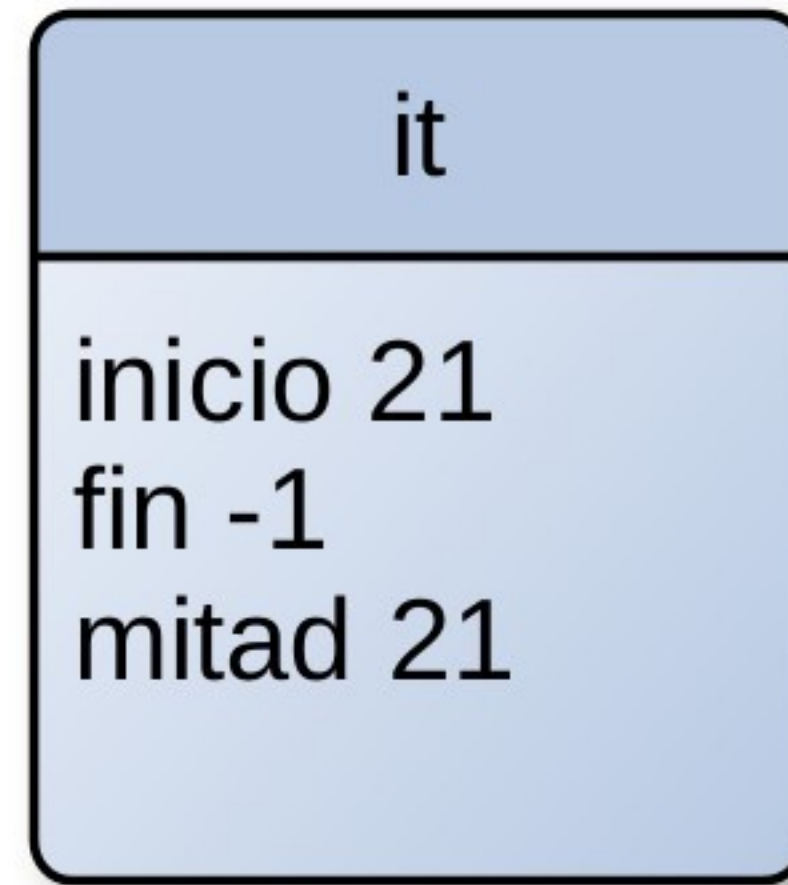
Inicio

Mitad

Final

21





7

2

1

-4

0

9

13

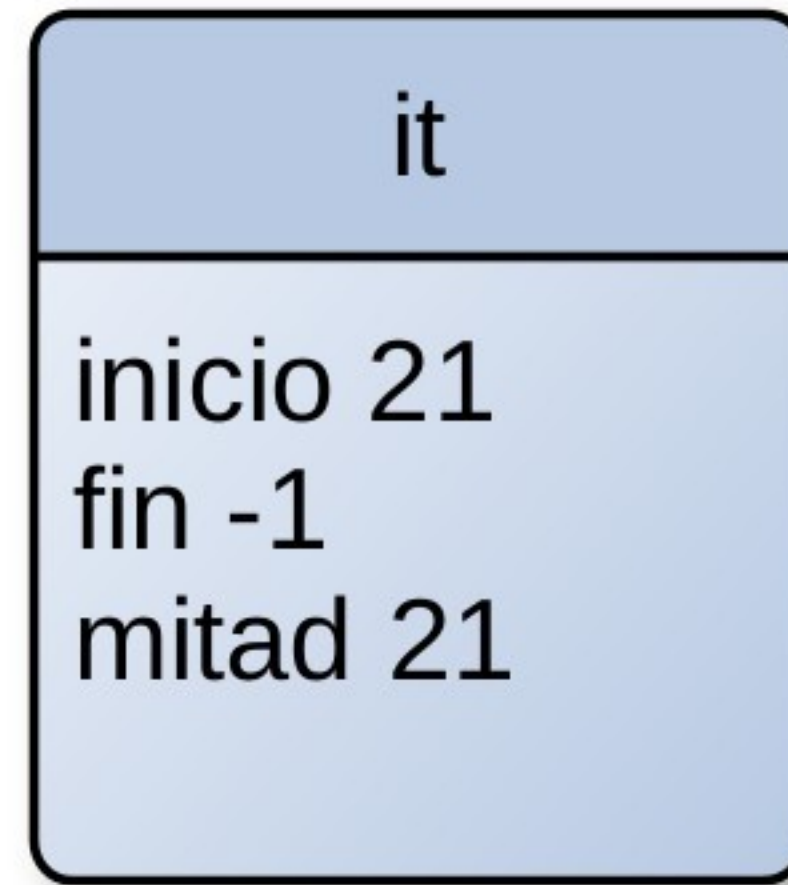
Inicio

Mitad

Final

21

-1



7

2

1

-4

0

9

Inicio

Mitad

13

21

Final

-1



7

2

1

-4

0

9

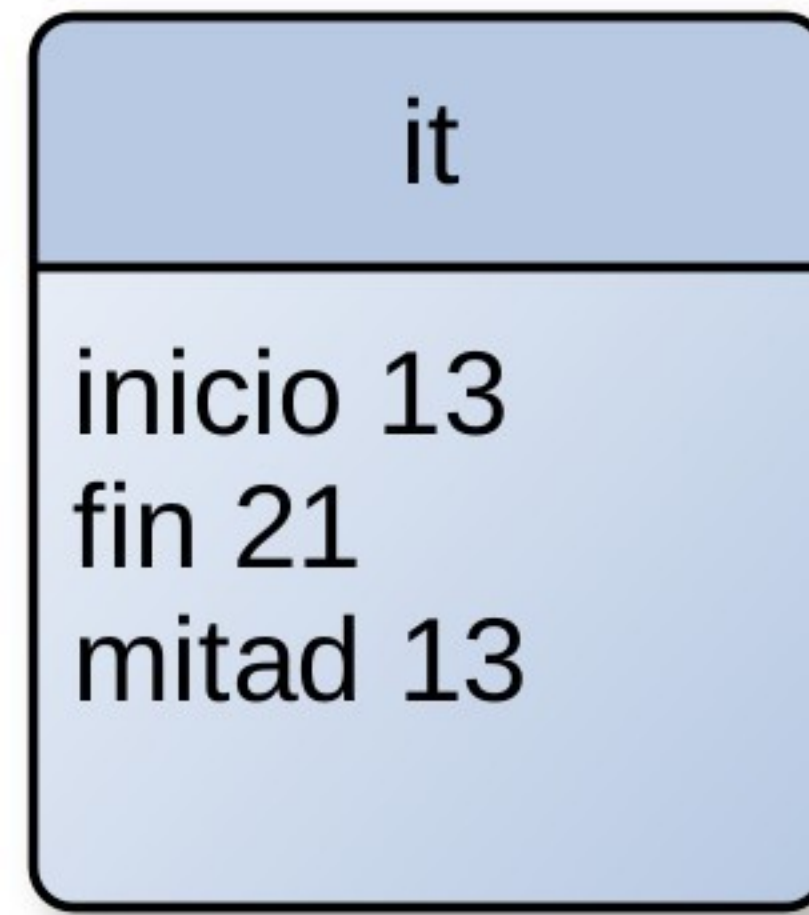
Inicio

Mitad

Final

13

21



7

2

1

-4

0

9

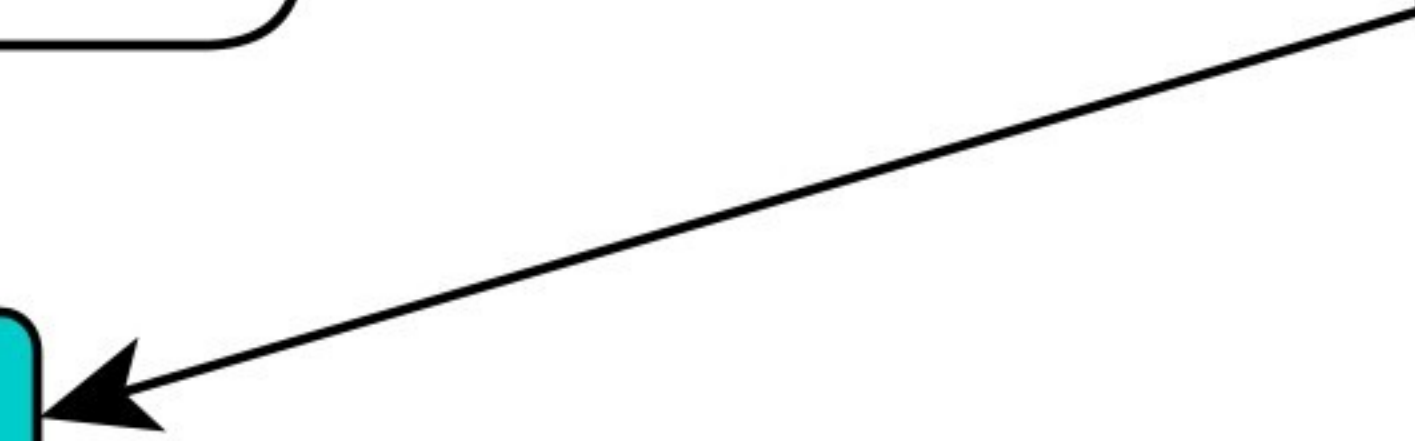
Inicio

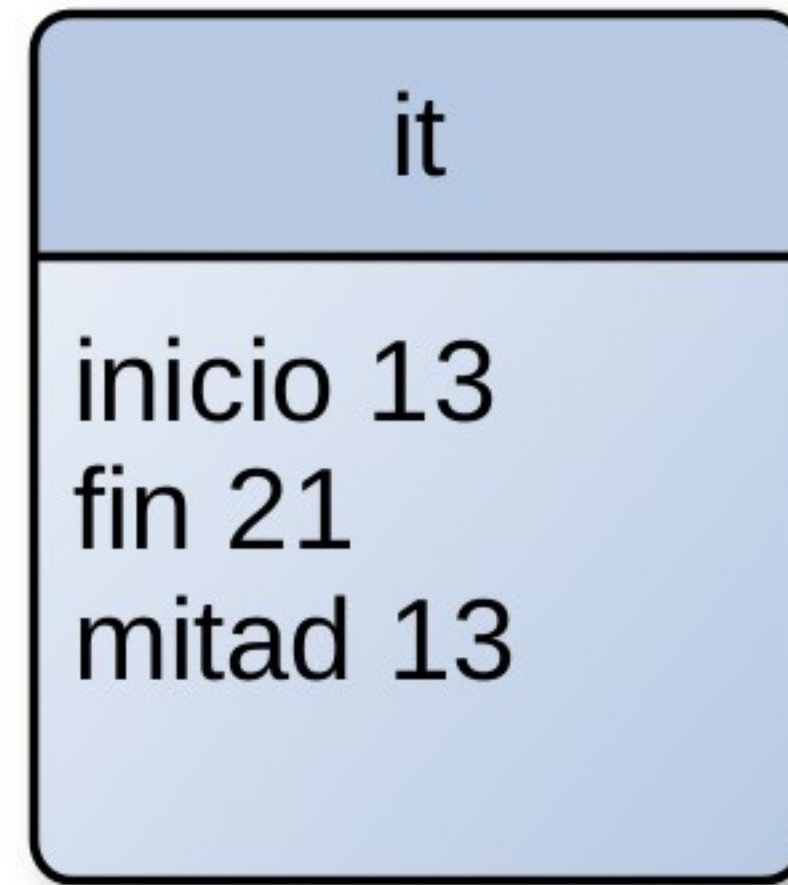
Mitad

Final

13

21





7

2

1

-4

0

9

Inicio

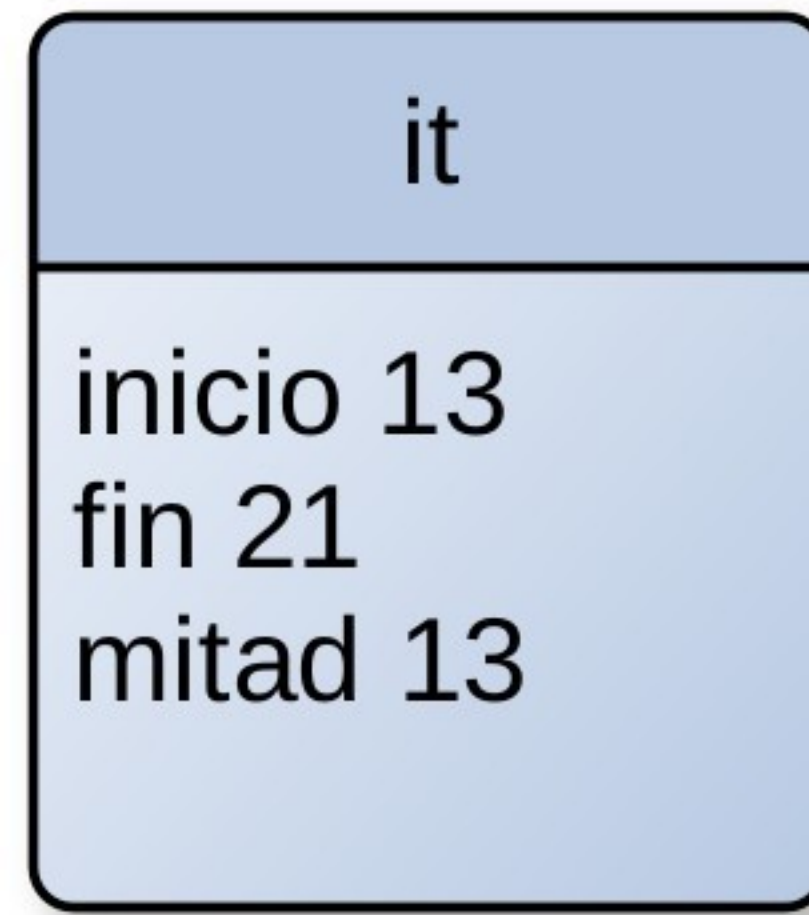
Mitad

Final

13

21

-1



7

2

1

-4

0

9

Inicio

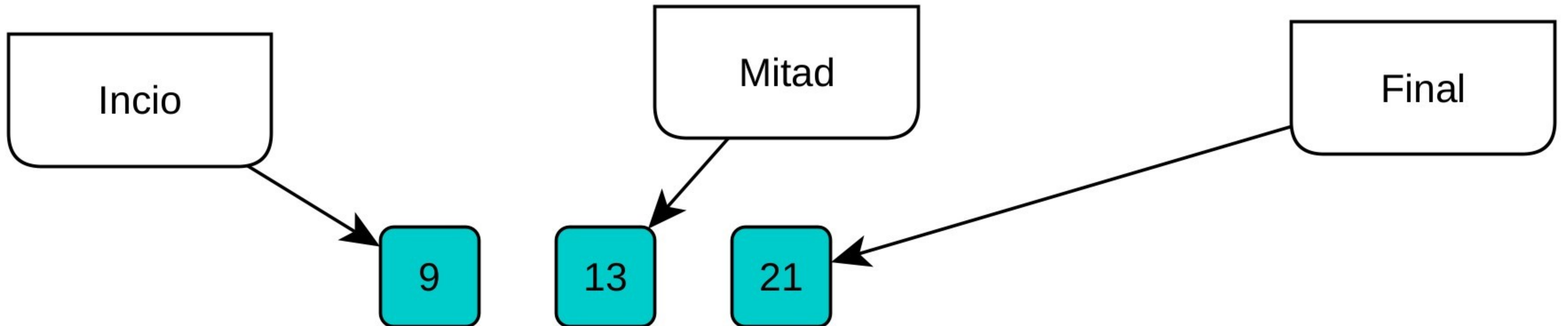
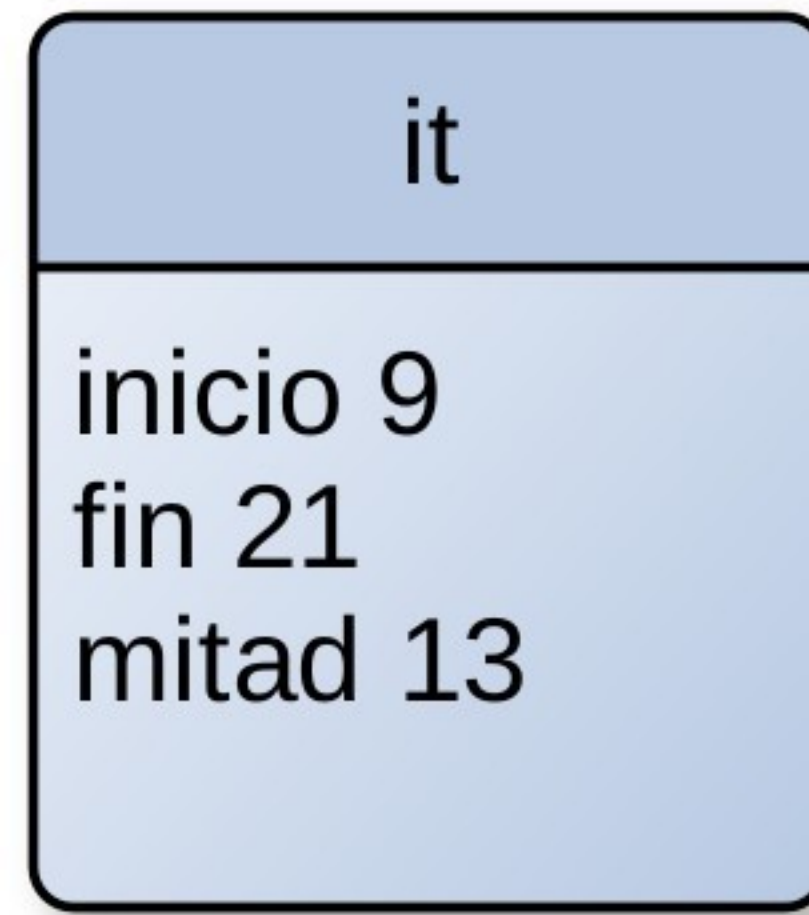
Mitad

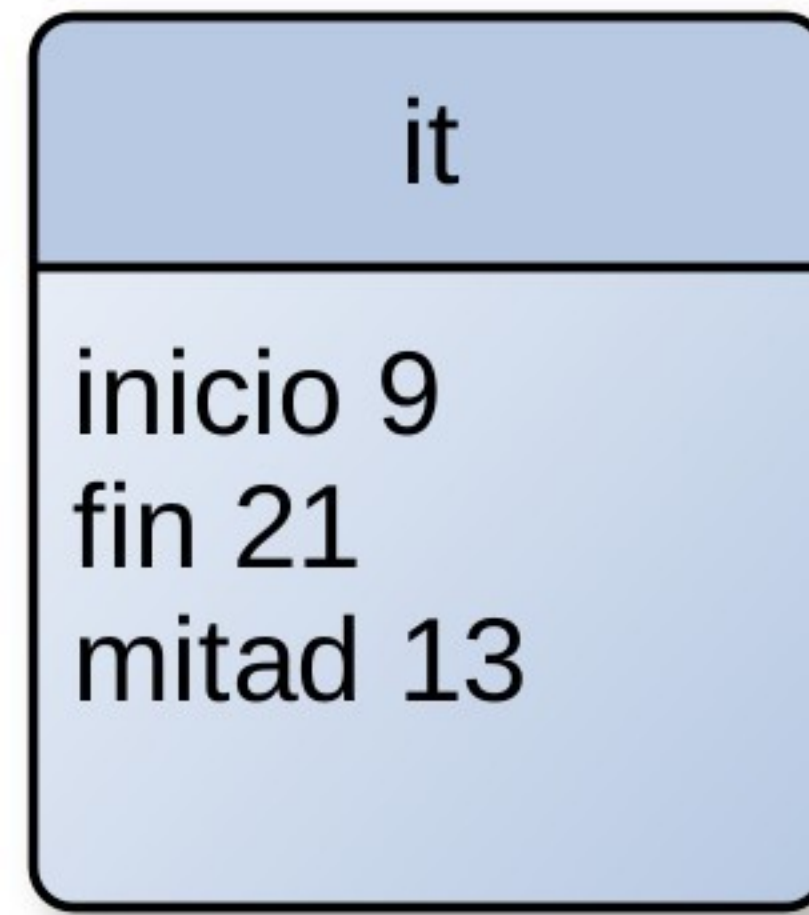
Final

13

21







7

2

1

-4

0

Inicio

Mitad

Final

9

13

21



7

2

1

-4

0

Inicio

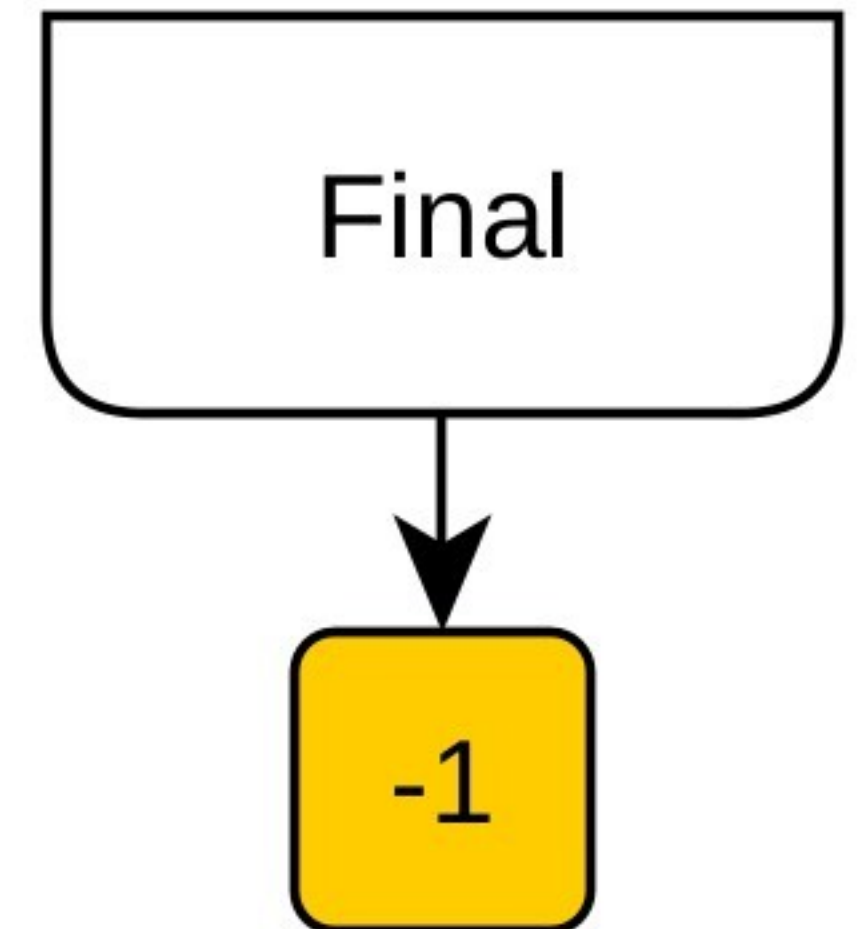
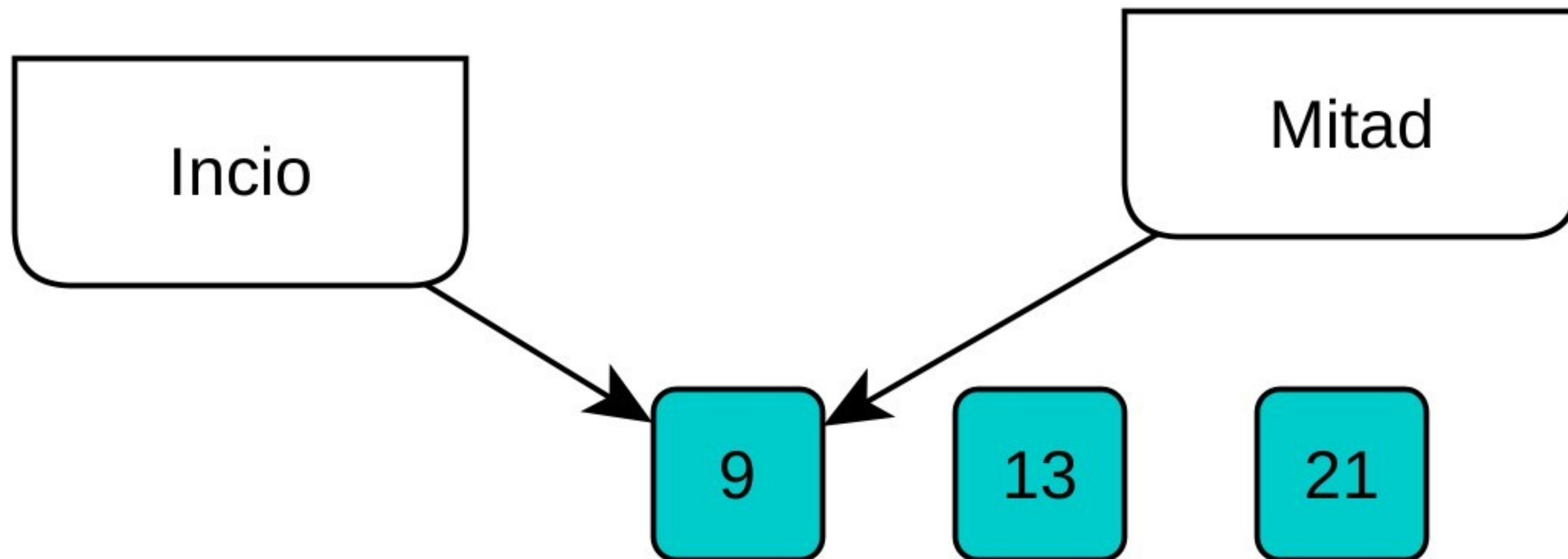
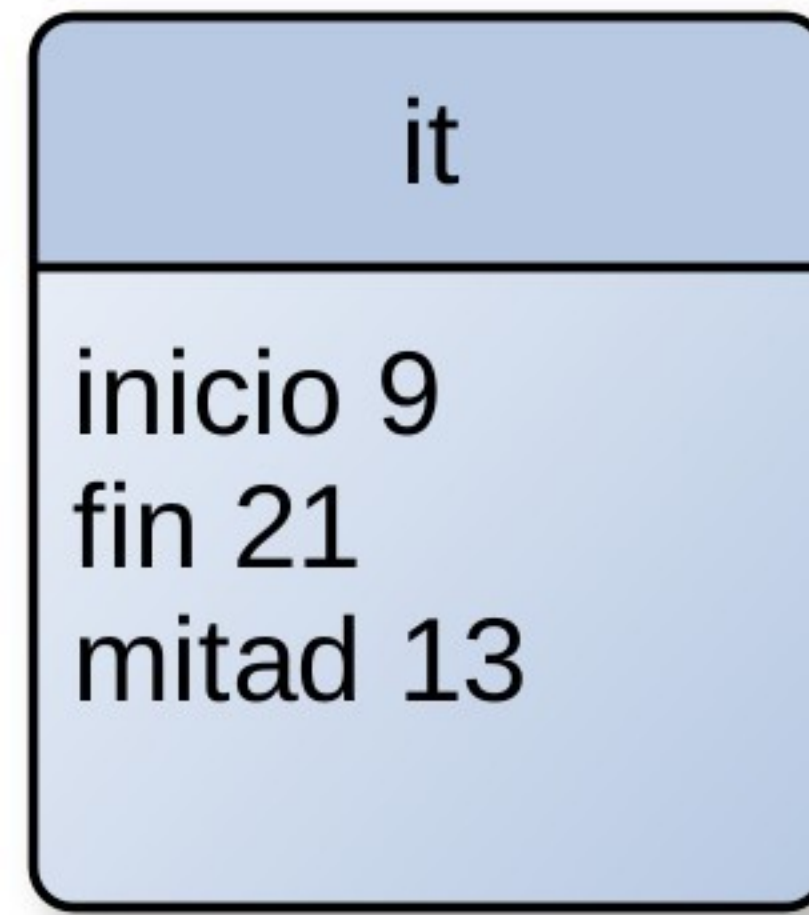
Mitad

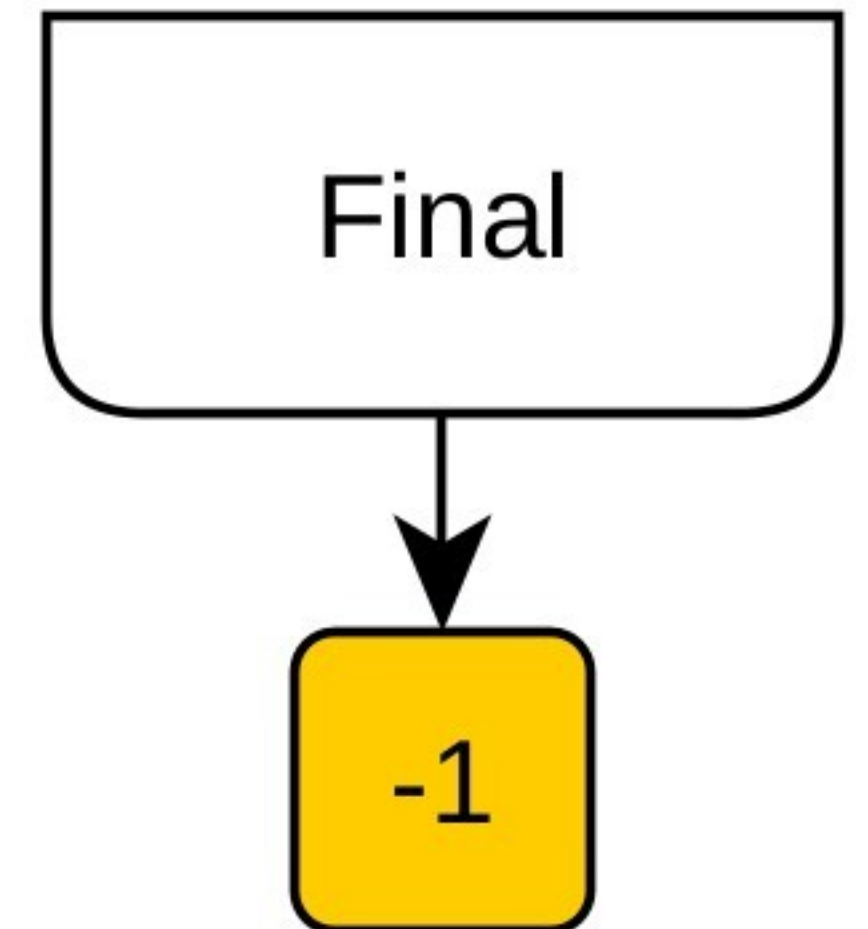
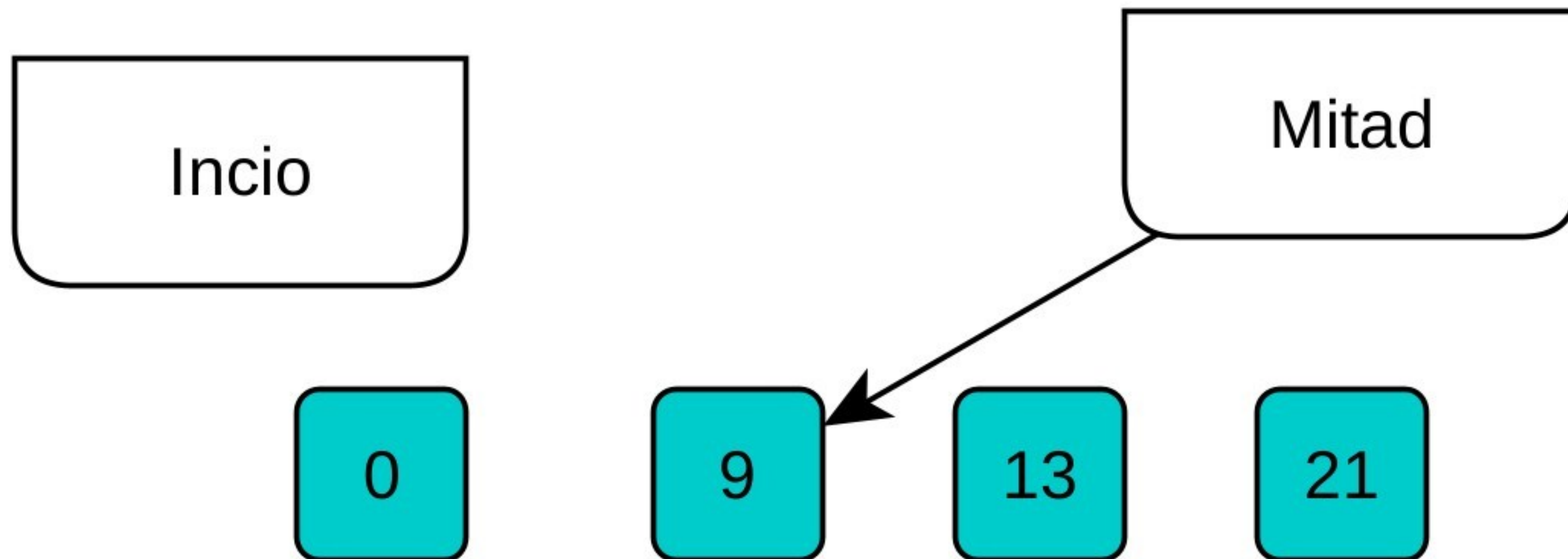
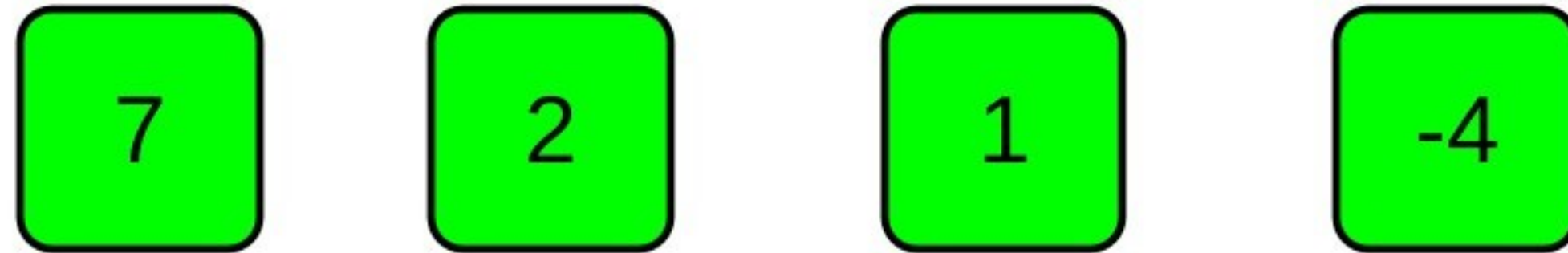
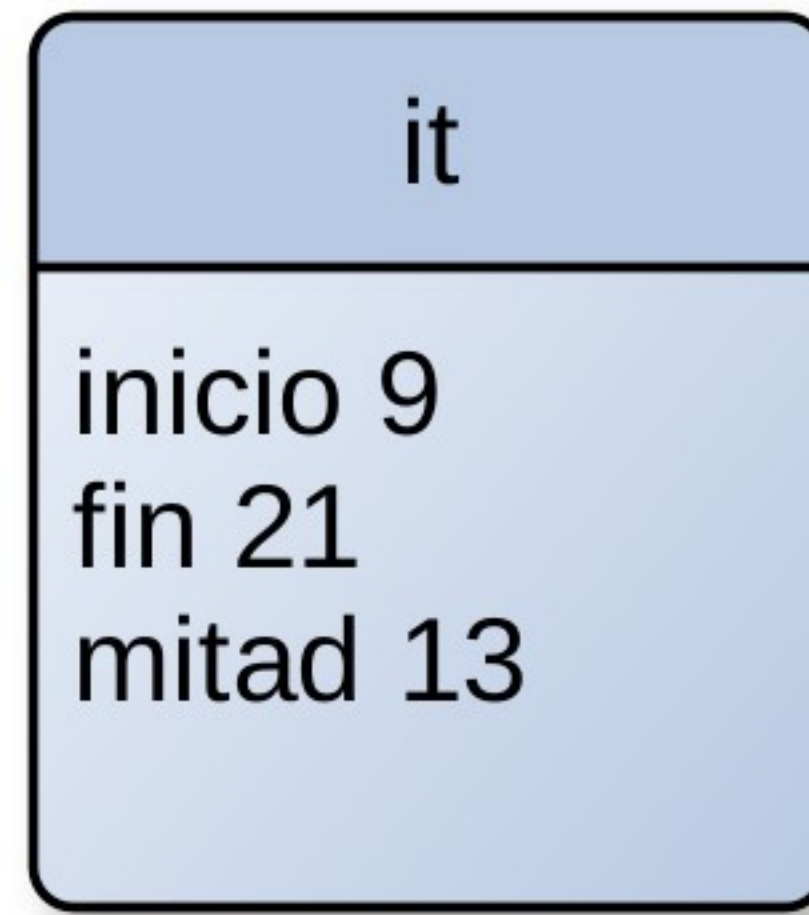
Final

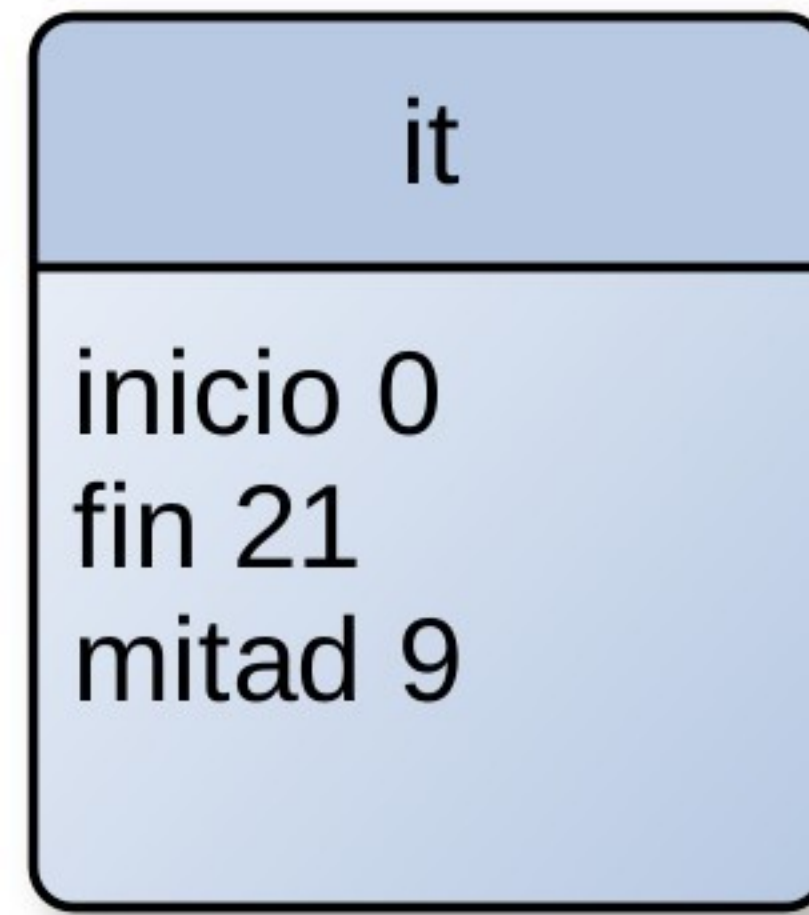
9

13

21







7

2

1

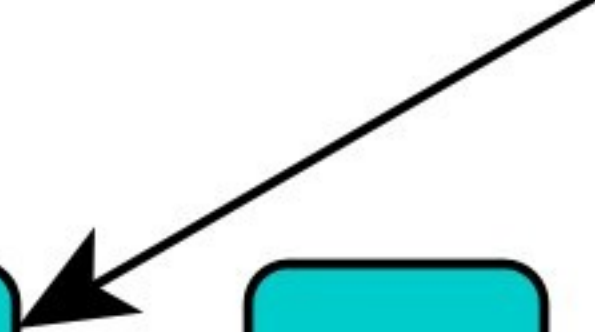
-4

Inicio



0

Mitad

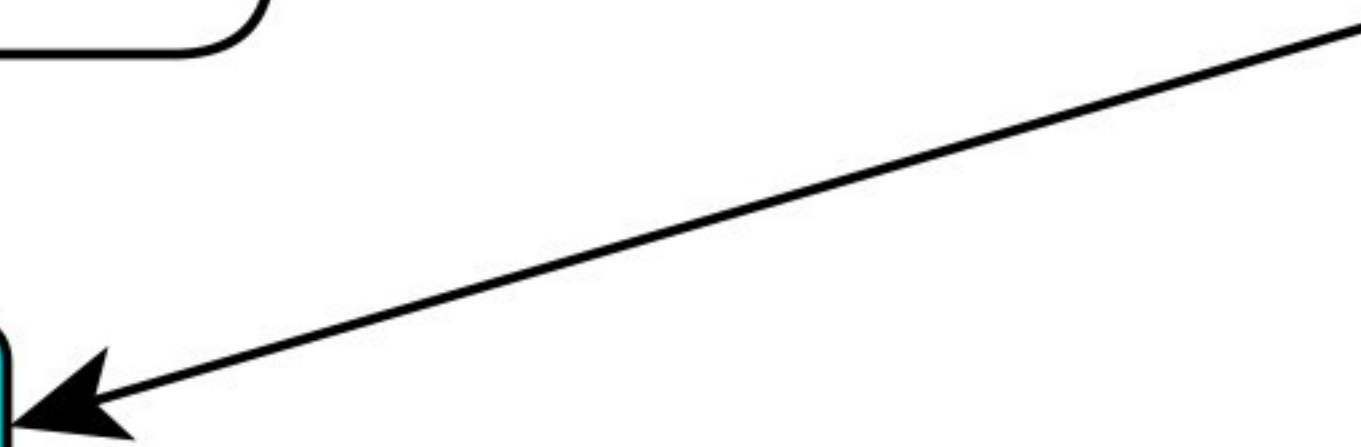


9

13

21

Final





7

2

1

-4

Inicio

Mitad

Final

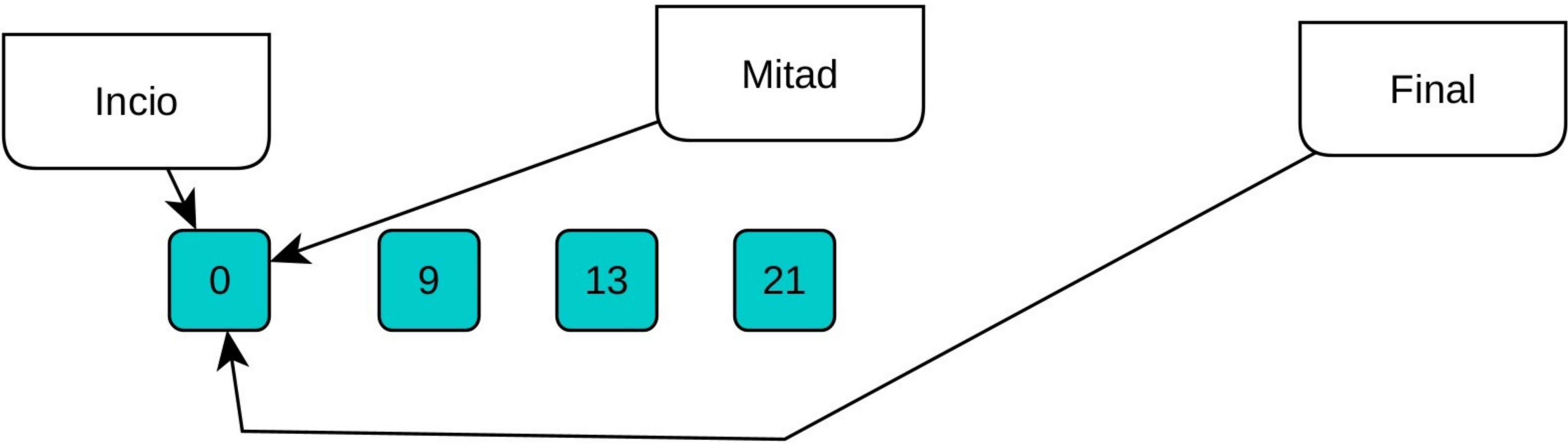
0

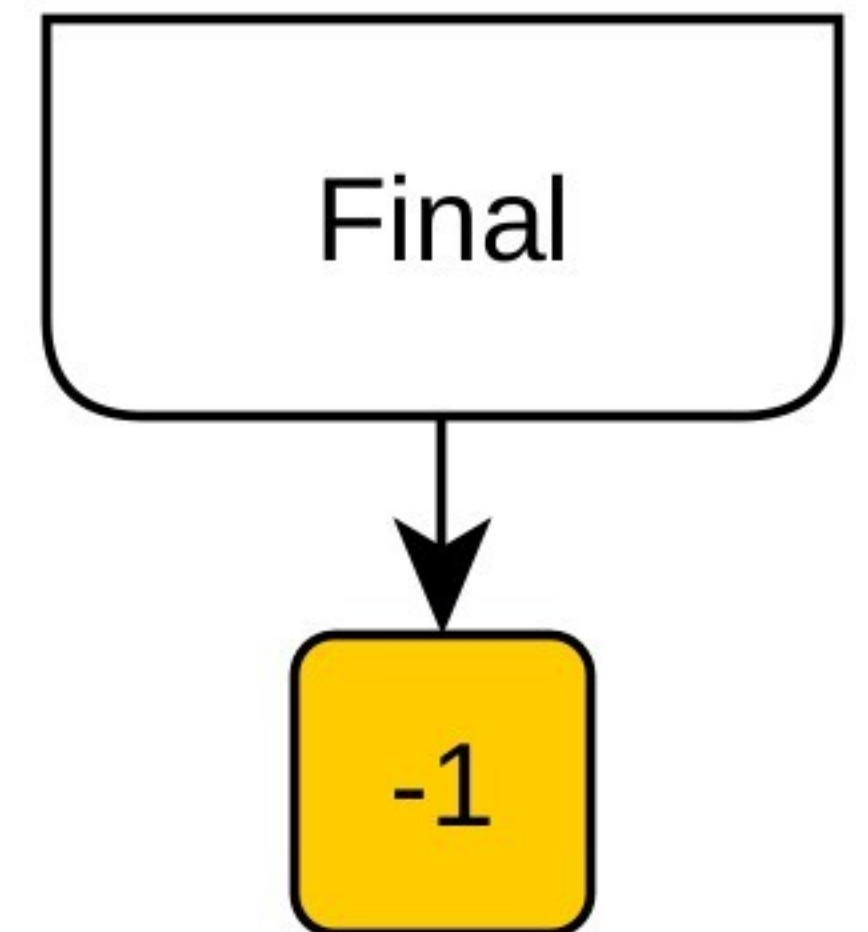
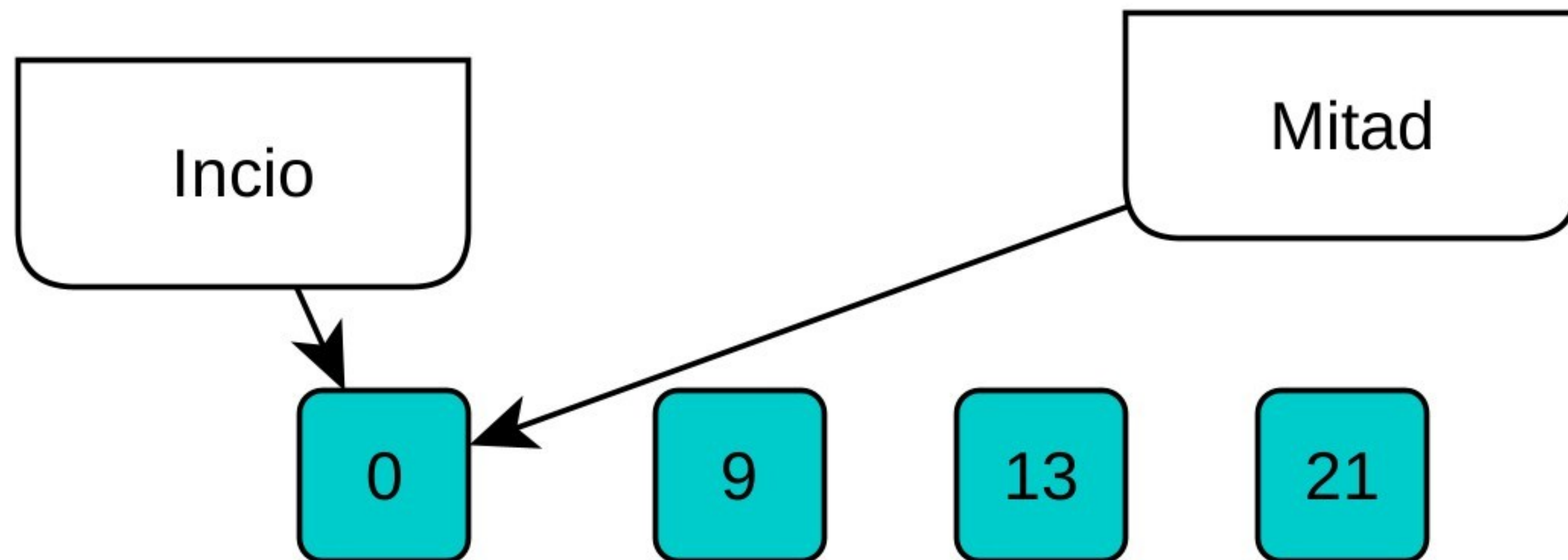
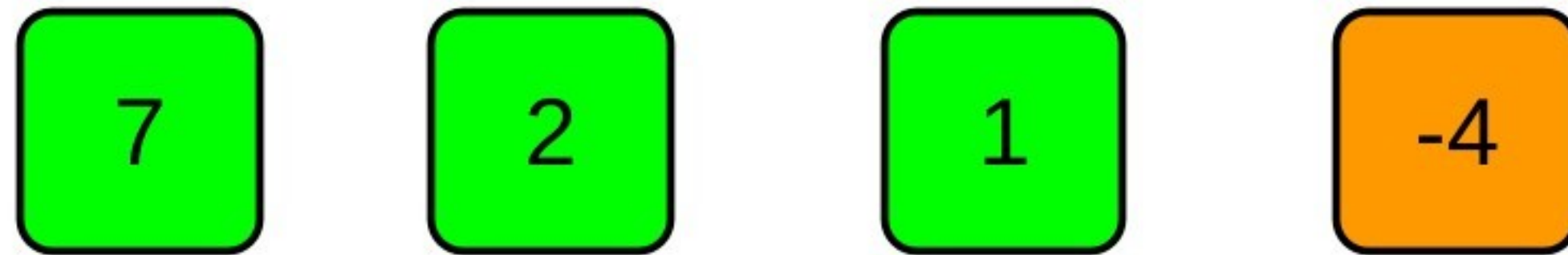
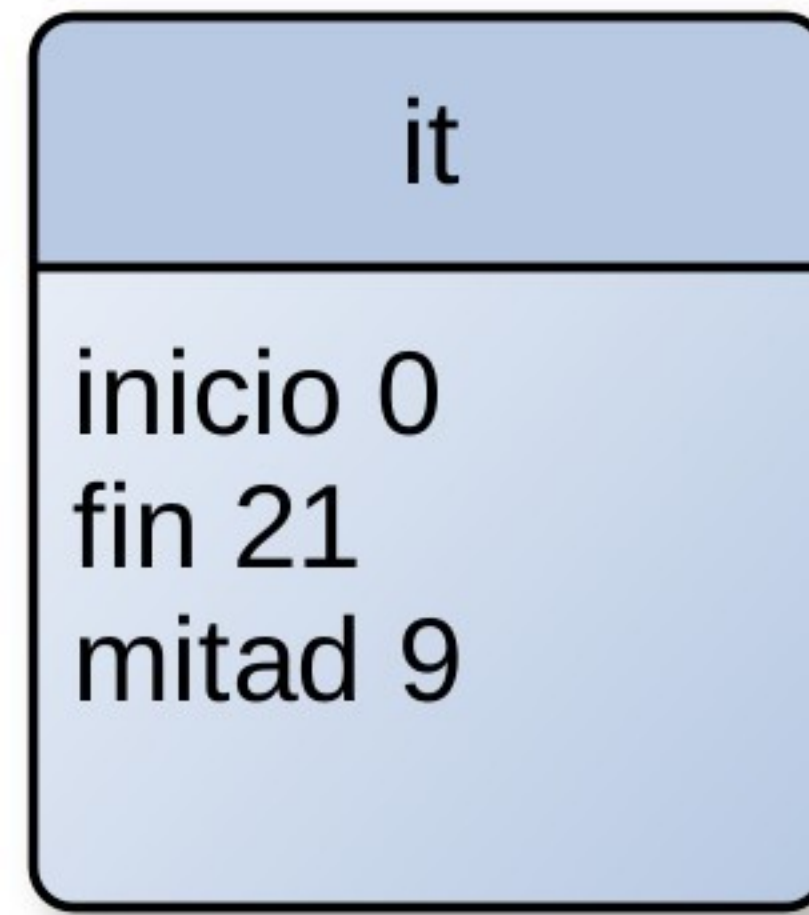
9

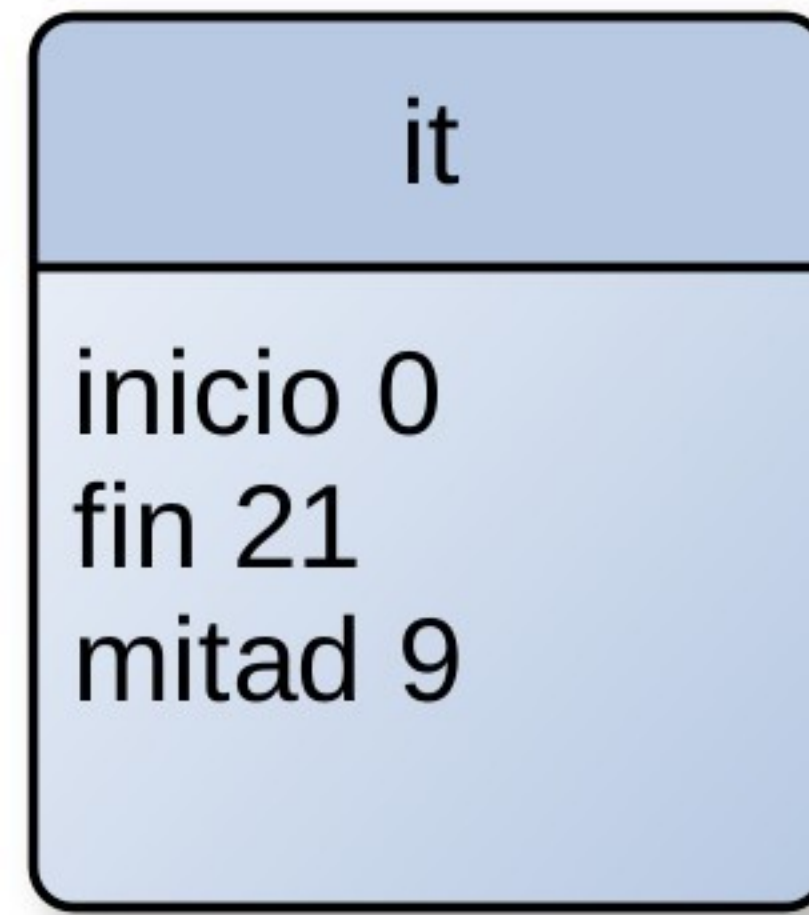
13

21









7

2

1

Inicio

Mitad

-4

0

9

13

21

Final



-1



7

2

1

Inicio

Mitad

Final

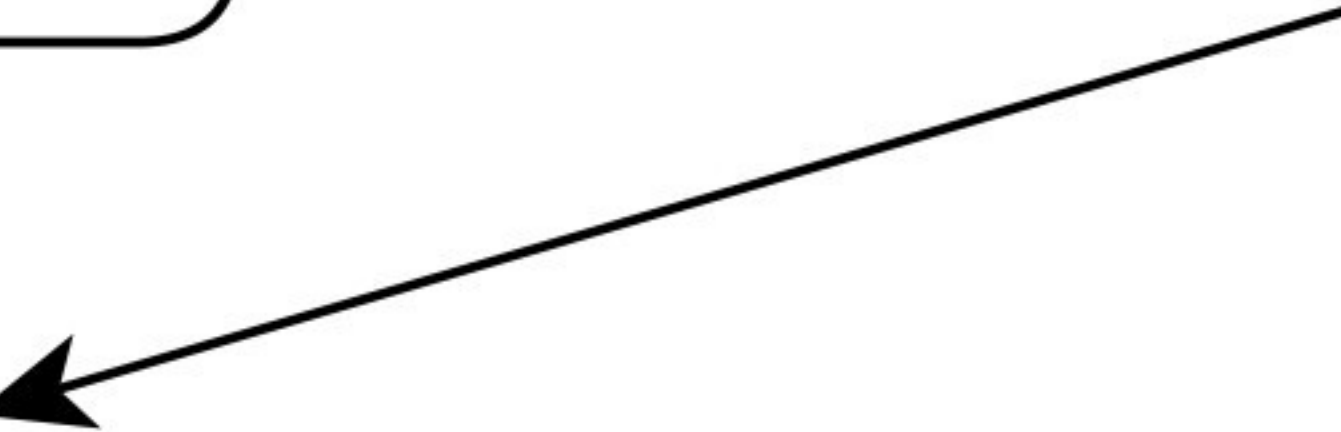
-4

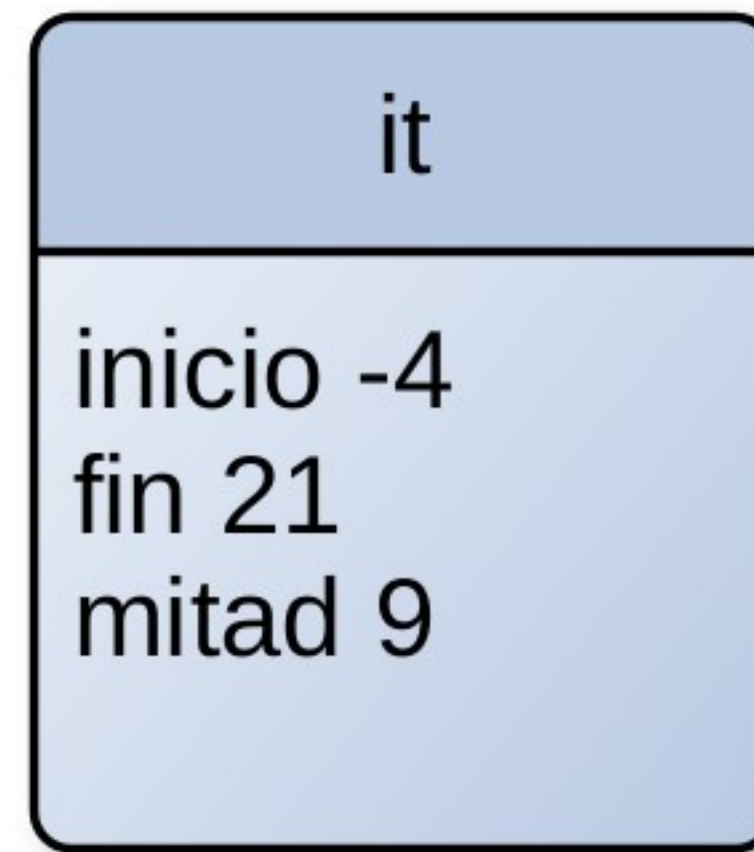
0

9

13

21





7

2

1

Inicio

Mitad

Final

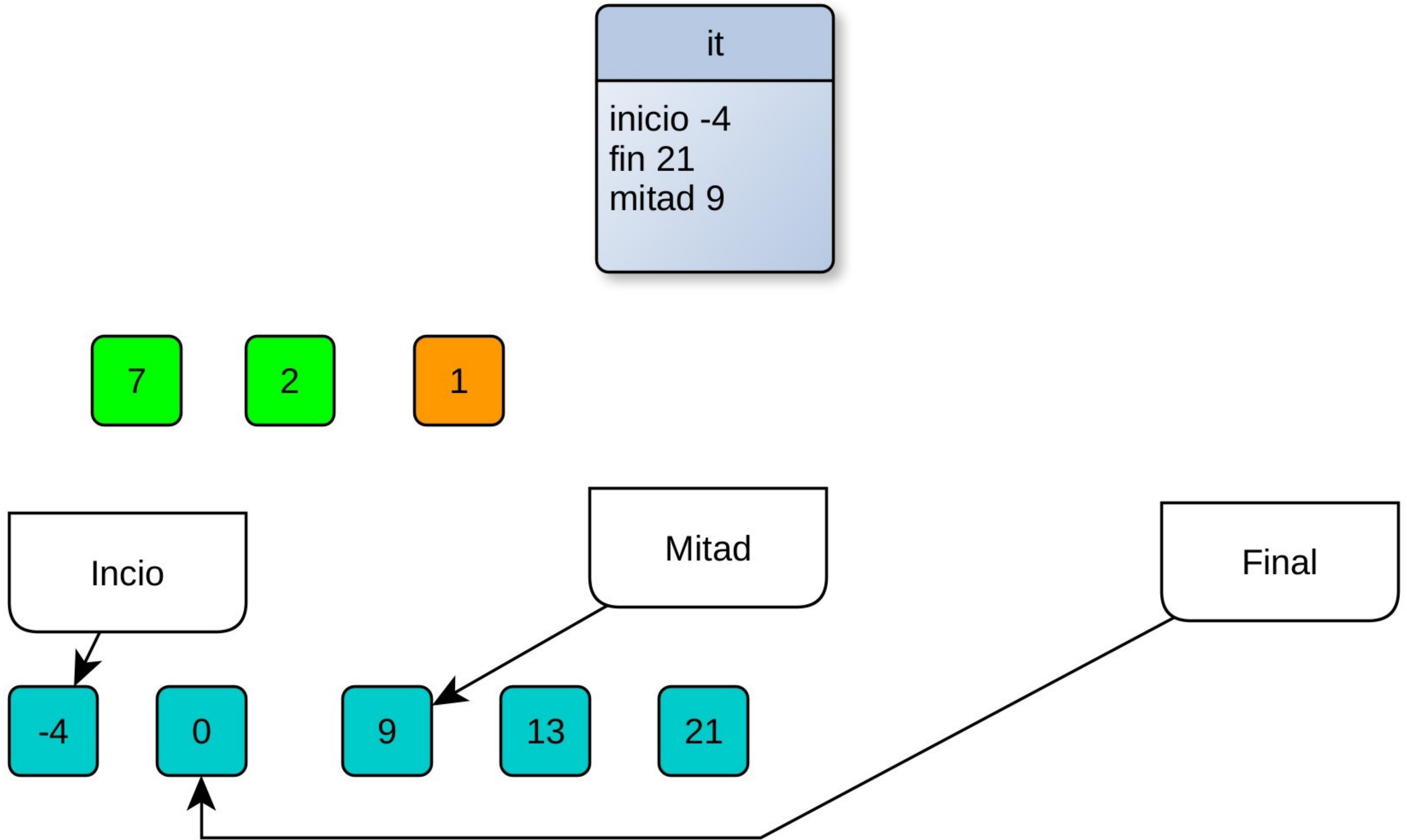
-4

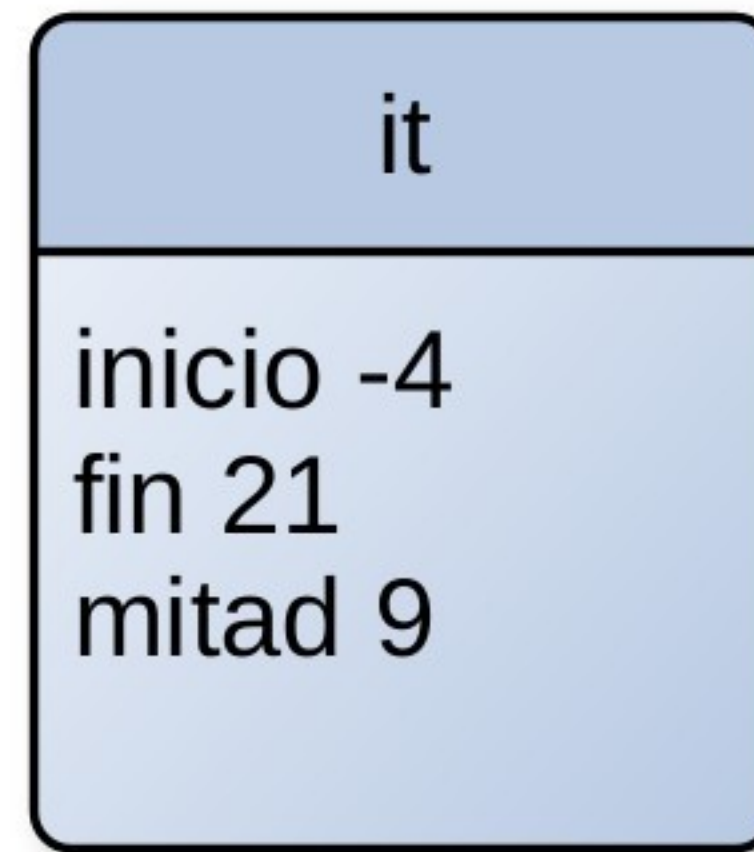
0

9

13

21





7

2

1

Inicio

Mitad

Final

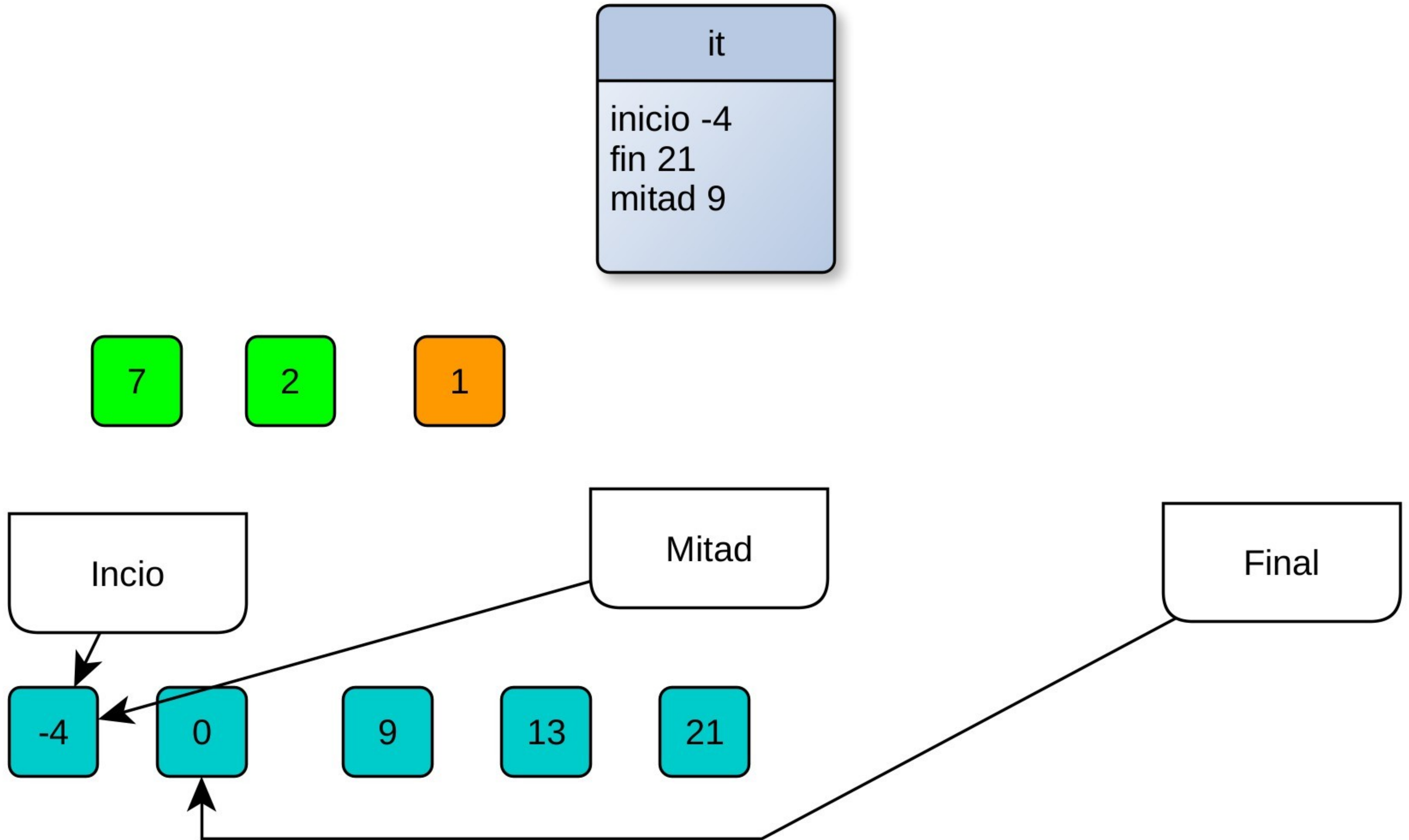
-4

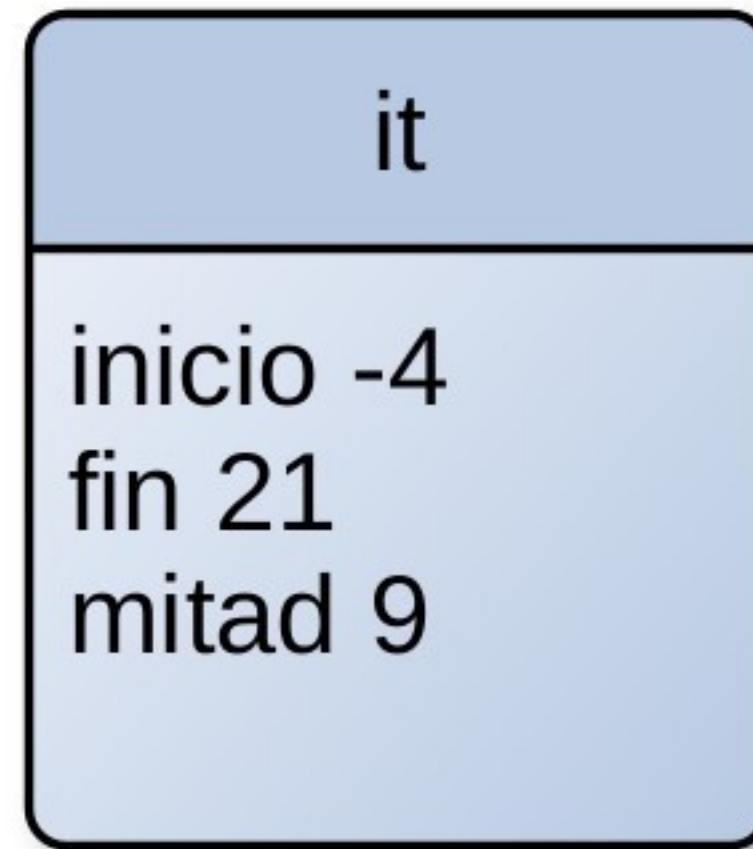
0

9

13

21





7

2

1

Inicio

Mitad

Final

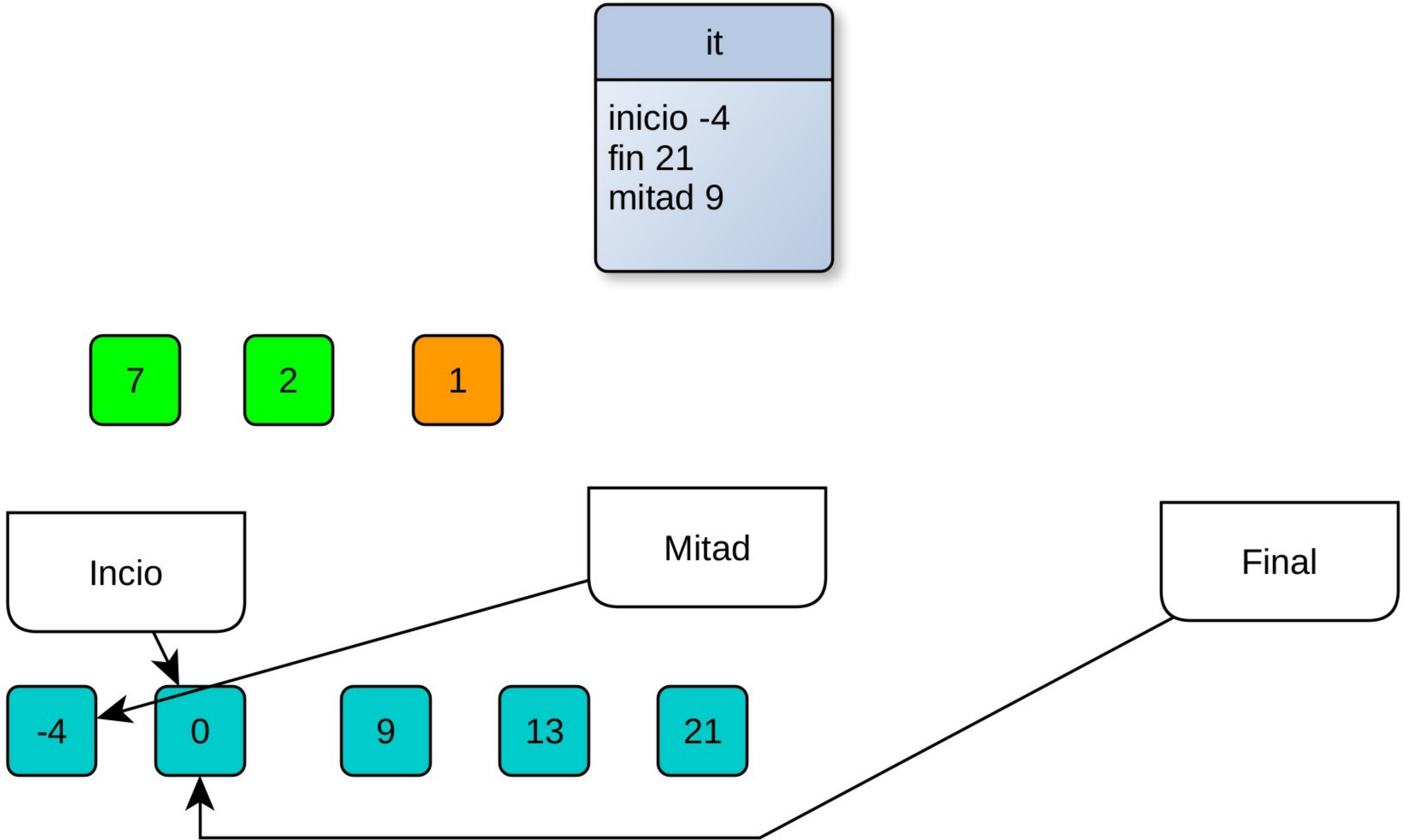
-4

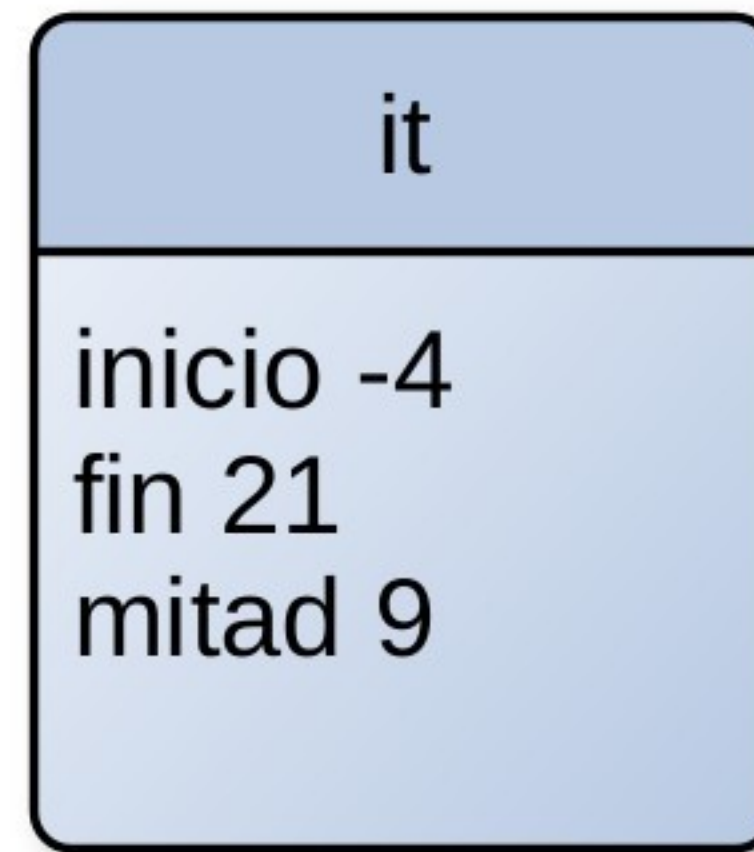
0

9

13

21





7

2

1

Inicio

Mitad

Final

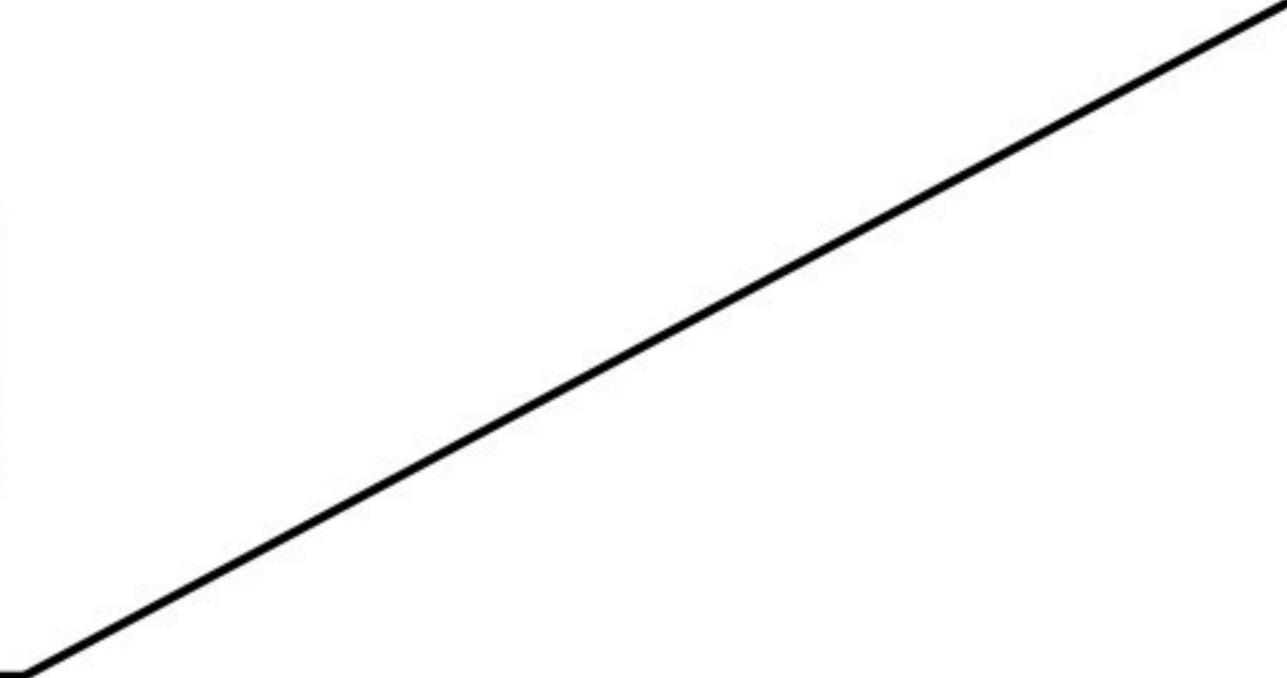
-4

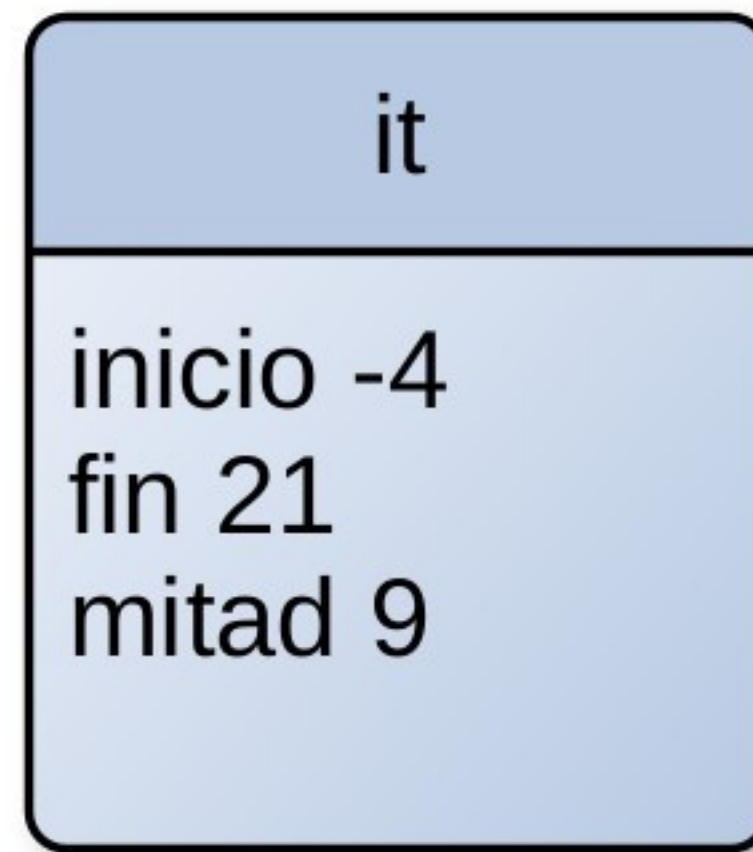
0

9

13

21





7

2

1

Inicio

Mitad

Final

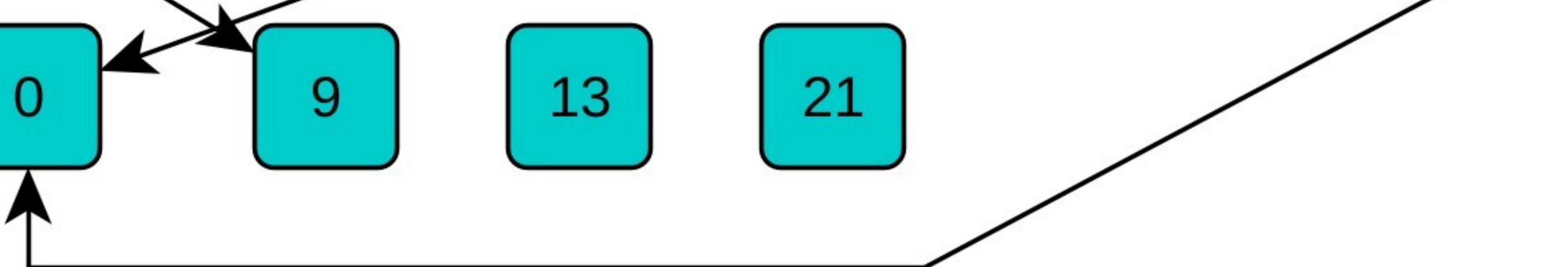
-4

0

9

13

21





7

2

1

Inicio

Mitad

Final

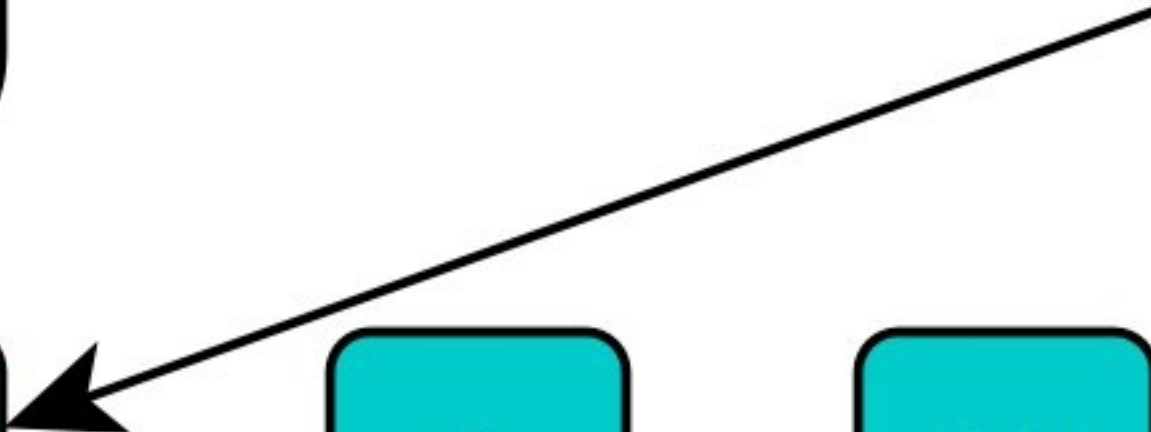
-4

0

9

13

21





7

2

Inicio

Mitad

Final

-4

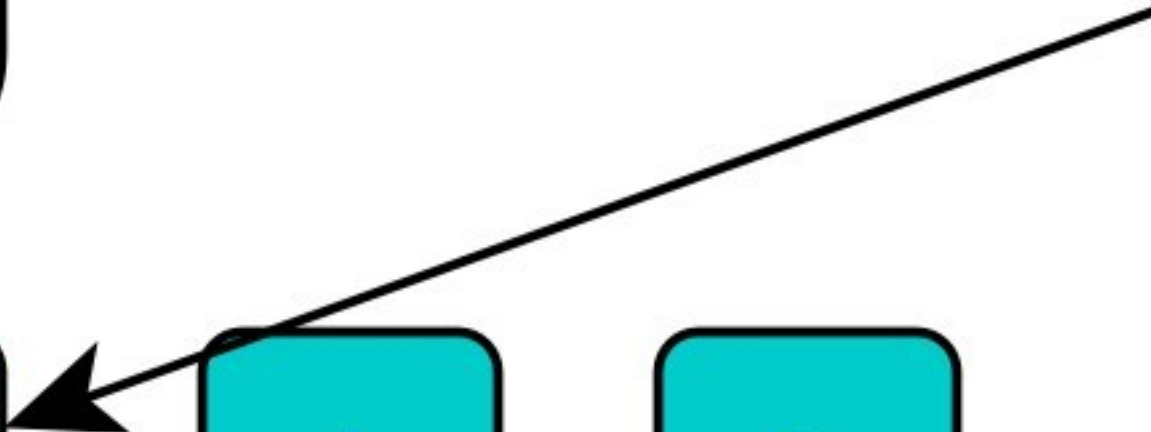
0

1

9

13

21





7

2

Inicio

Mitad

Final

-4

0

1

9

13

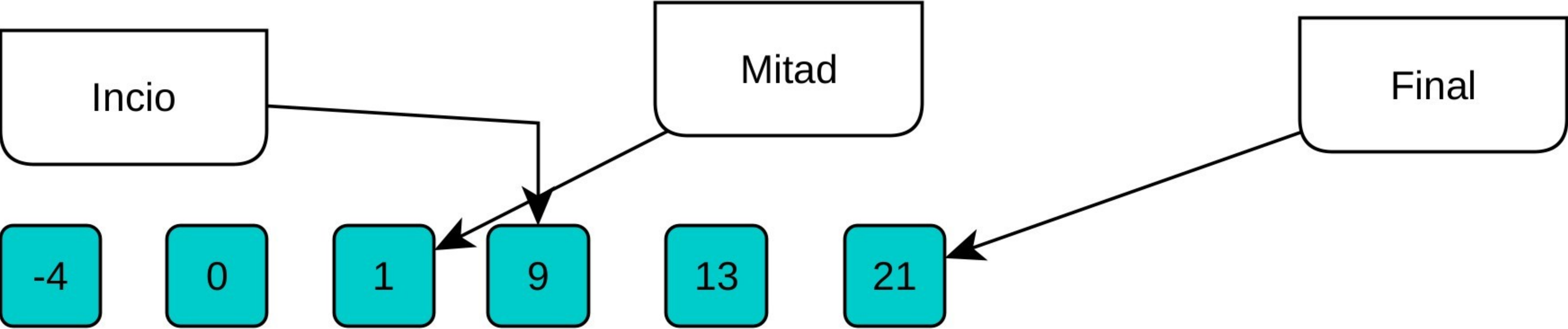
21

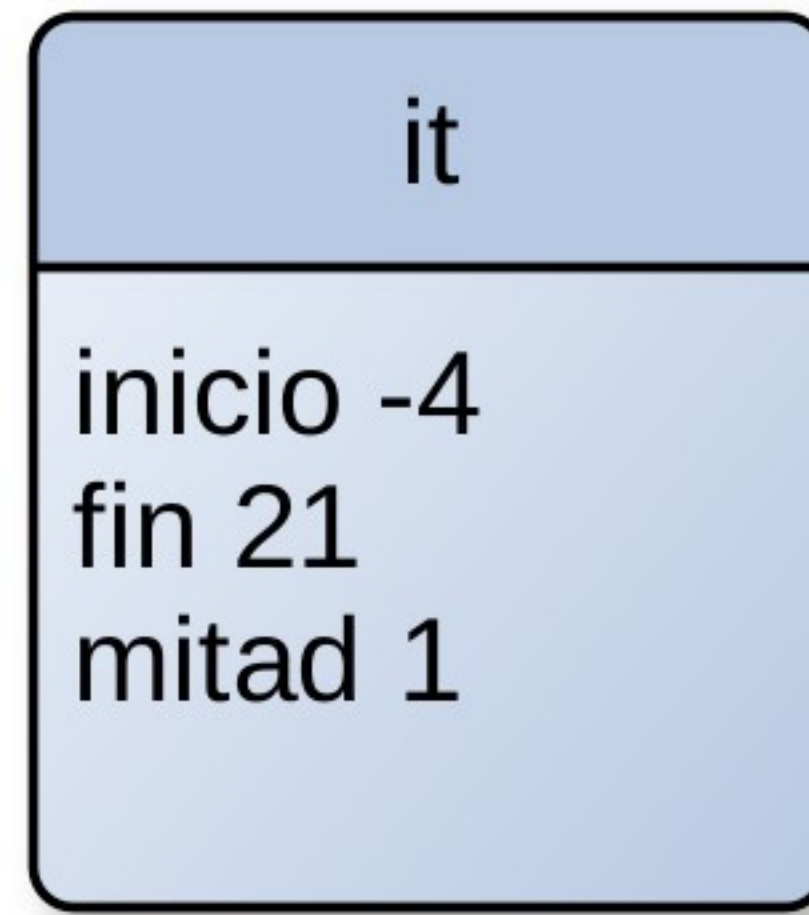




7

2





7

2

Inicio

Mitad

Final

-4

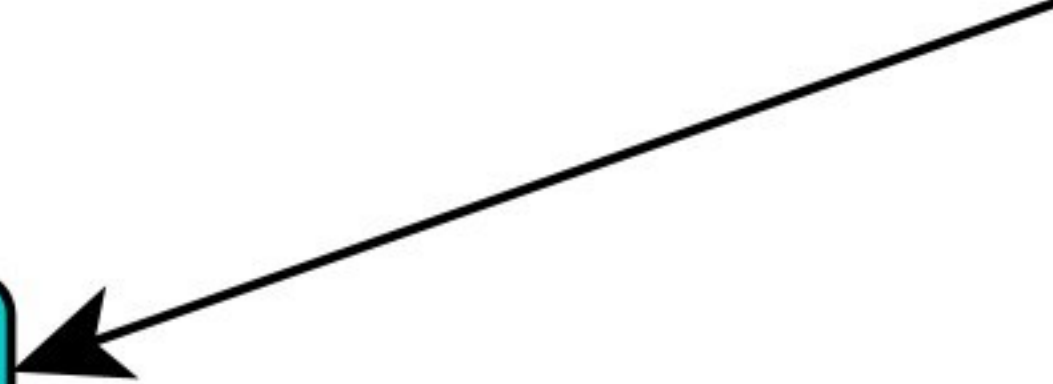
0

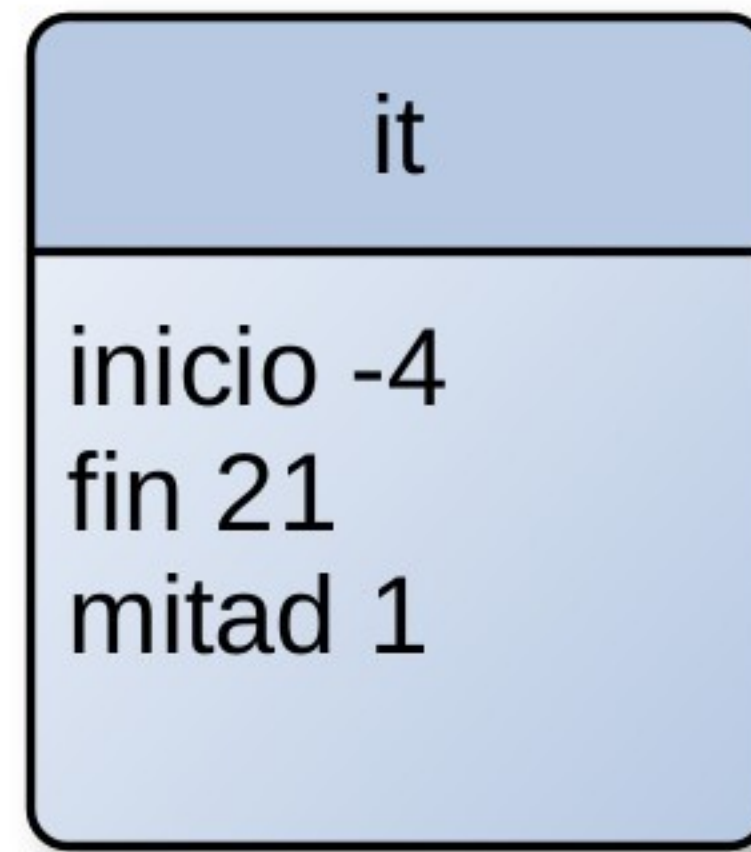
1

9

13

21





7

2

Inicio

Mitad

Final

-4

0

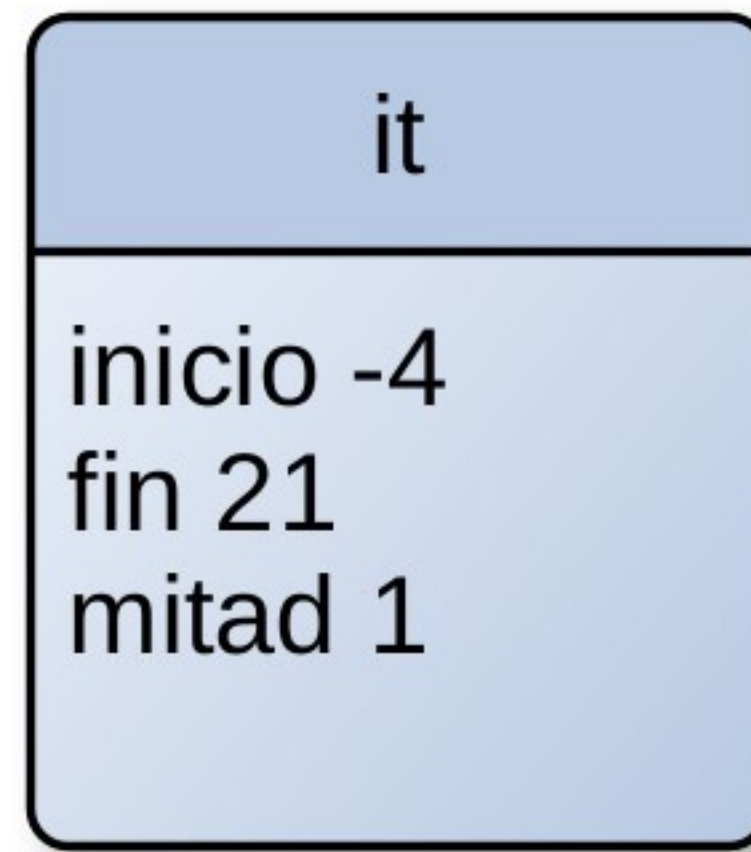
1

9

13

21





7

2

Inicio

Mitad

Final

-4

0

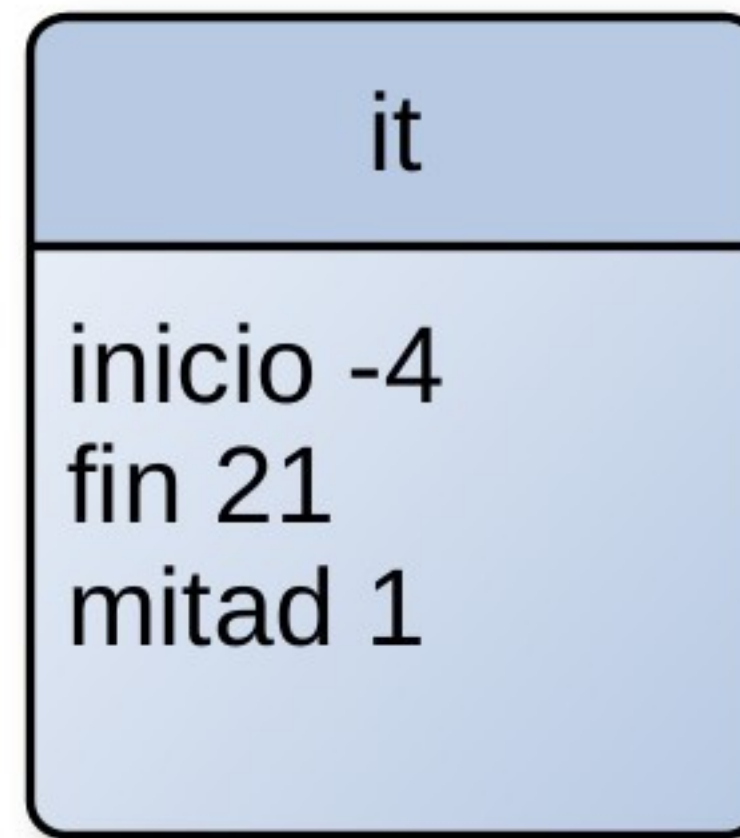
1

9

13

21





7

2

Inicio

Mitad

Final

-4

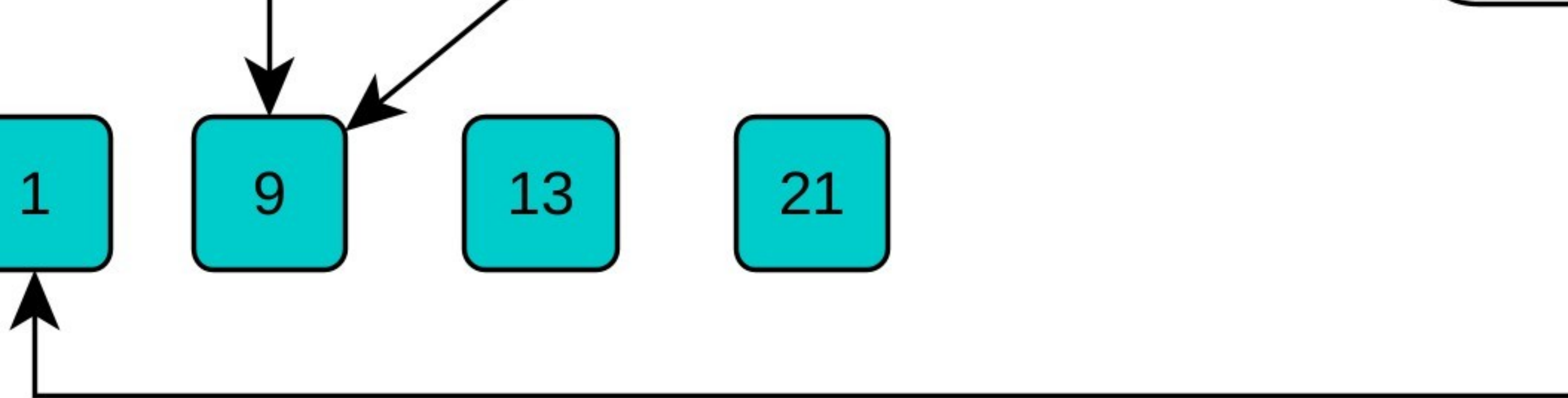
0

1

9

13

21





7

2

Inicio

Mitad

Final

-4

0

1

9

13

21





7

Inicio

Mitad

Final

-4

0

1

2

9

13

21



7

Inicio

-4

0

Mitad

1

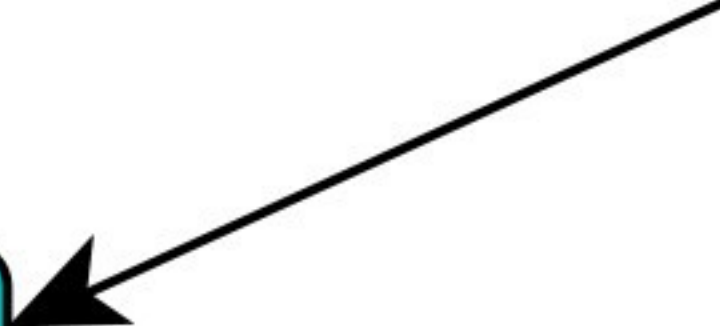
2

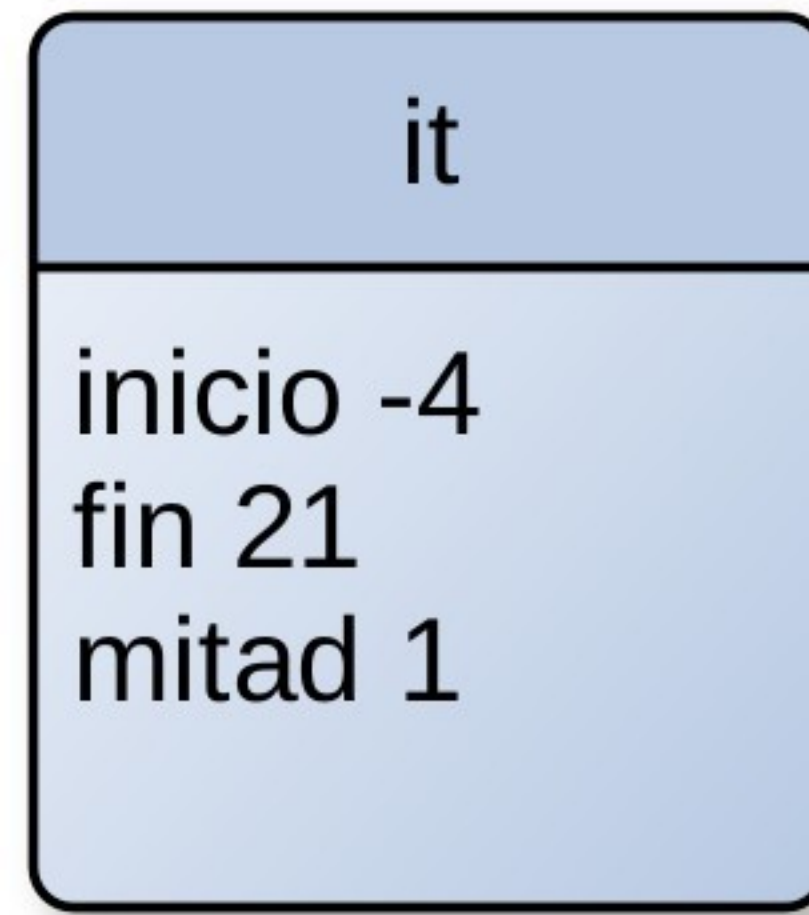
9

13

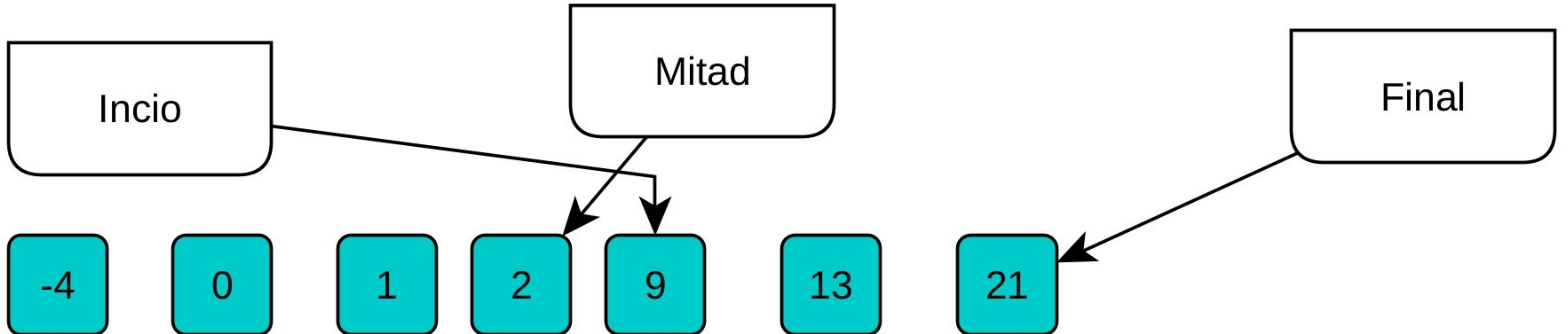
Final

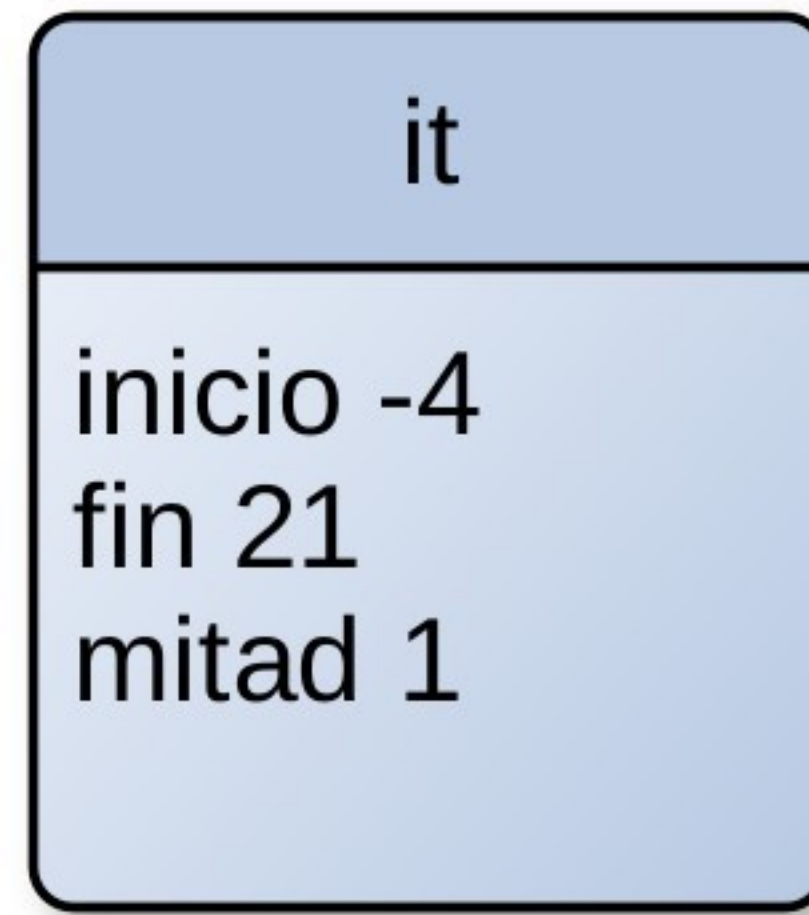
21



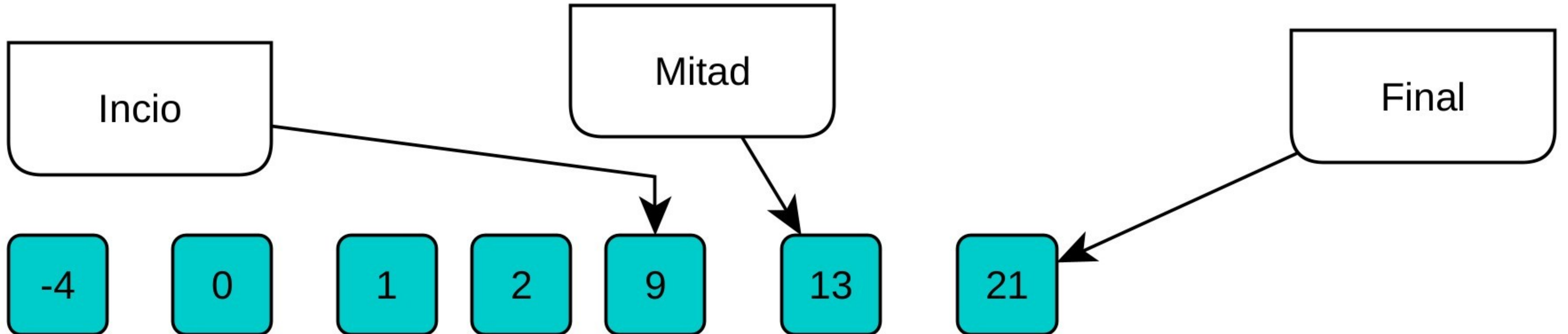


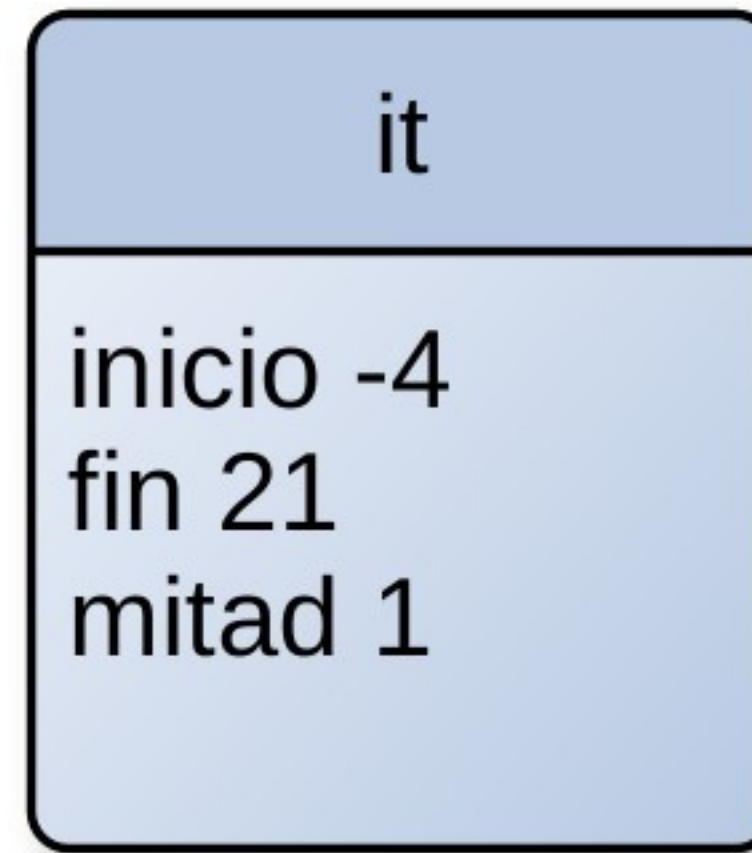
7



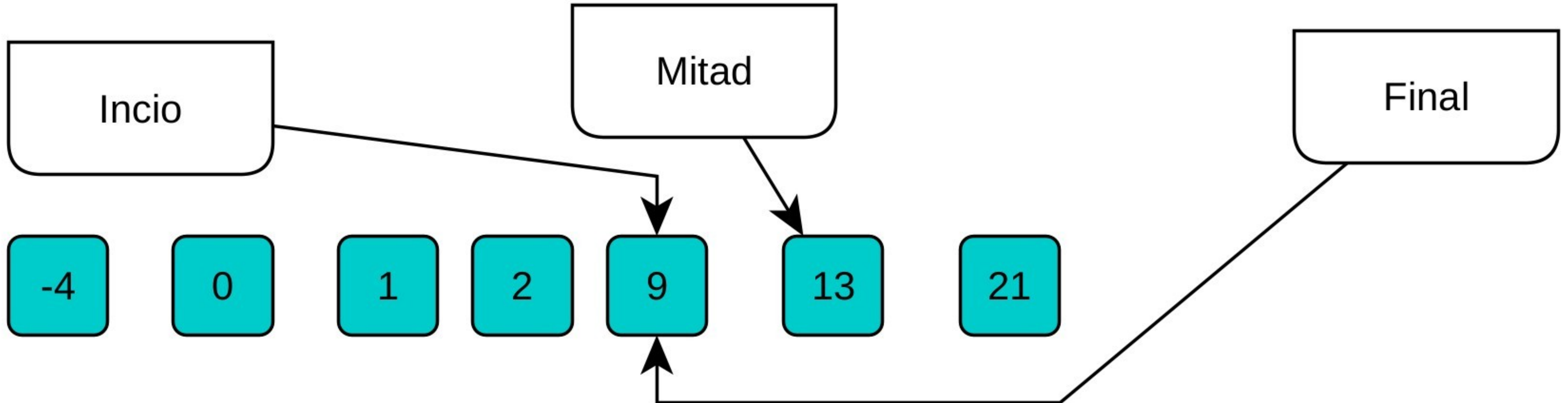


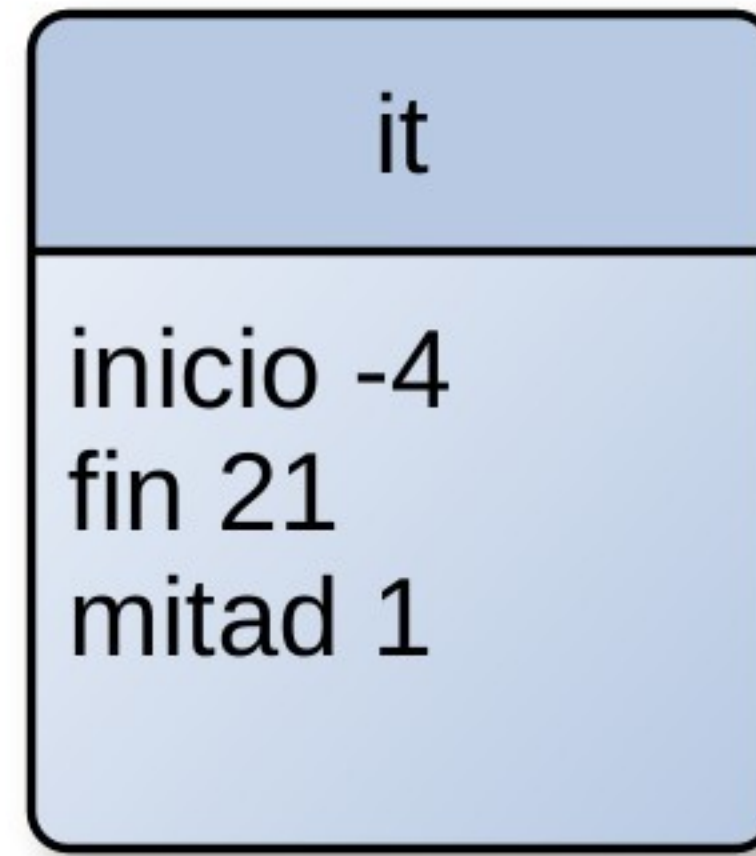
7



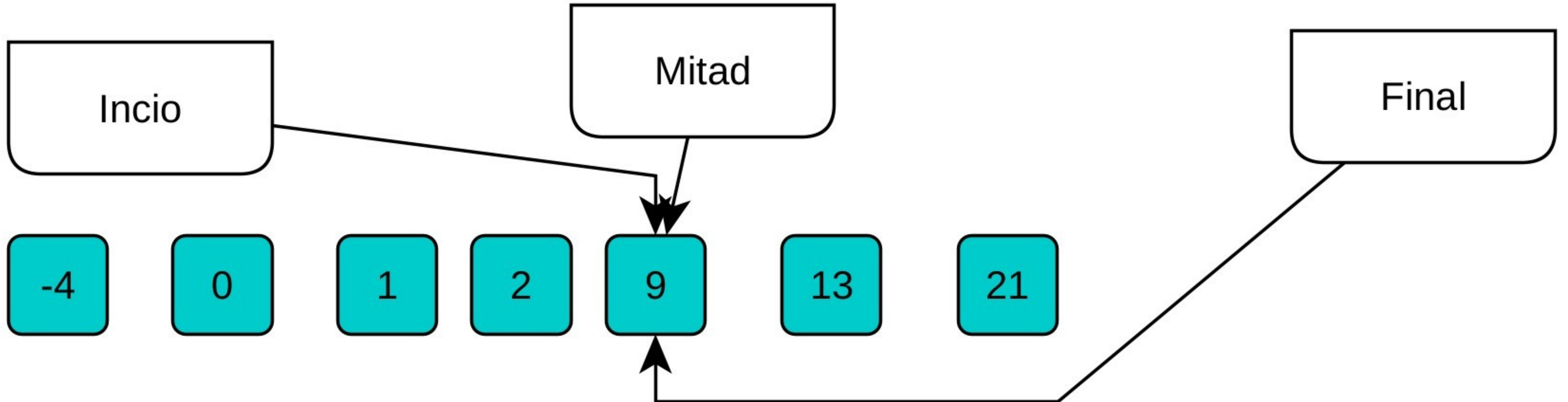


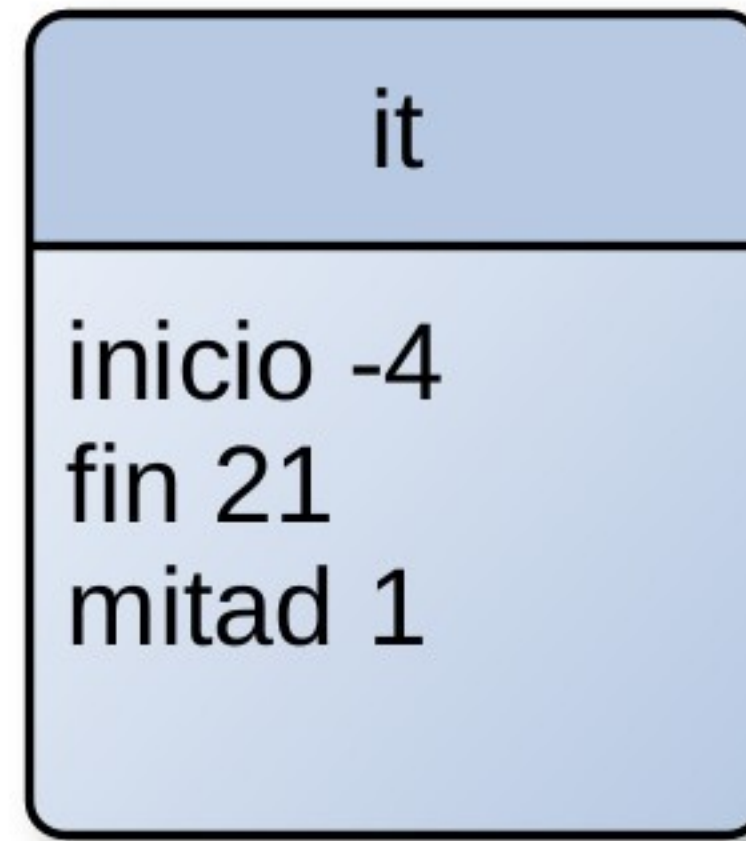
7



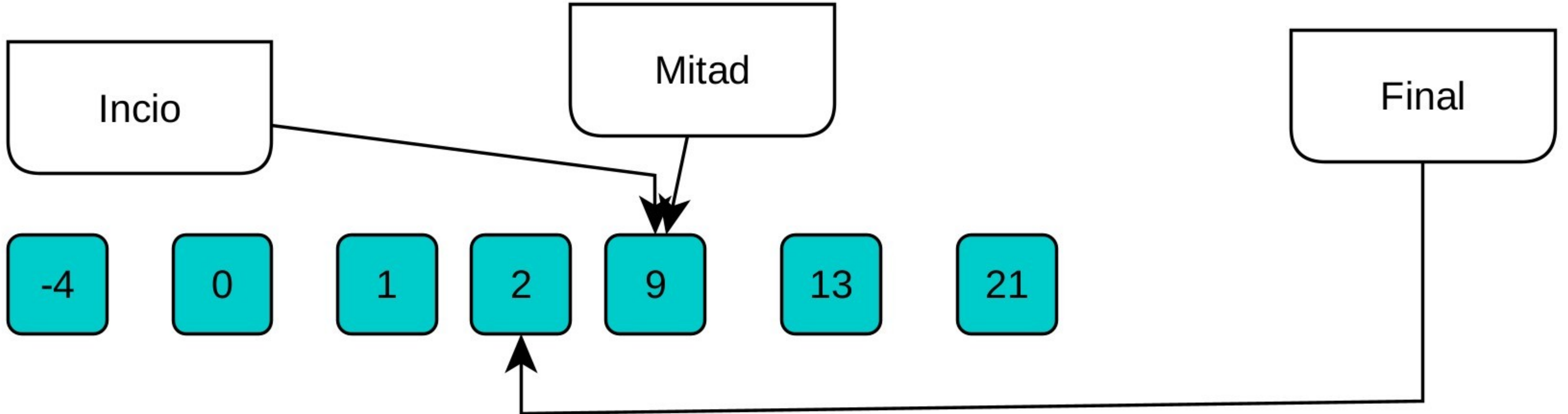


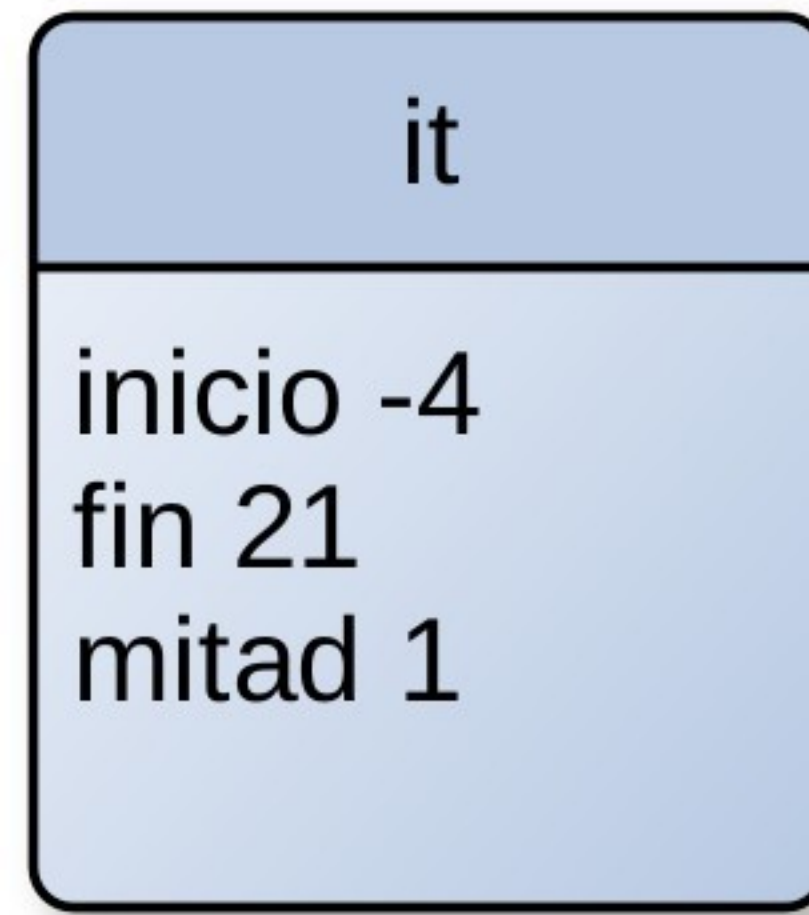
7





7





7

Inicio

Mitad

Final

-4

0

1

2

9

13

21

