CSC215

Math and Computer Science

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

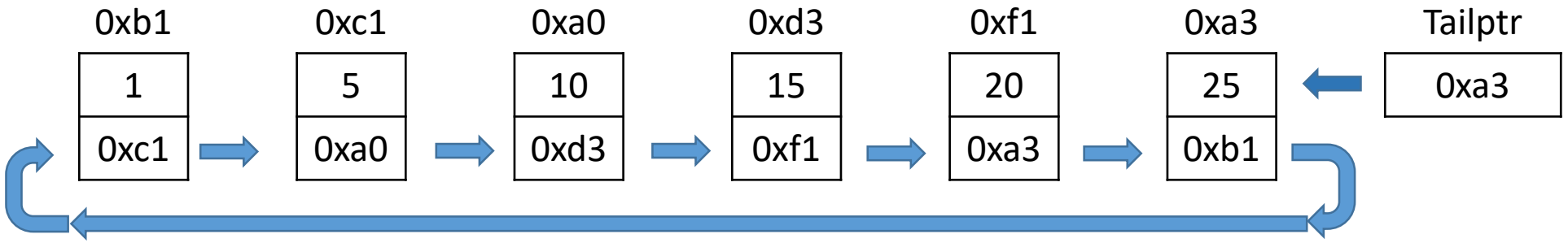
Circular Linked List

- Allows endless traversal if at least one item in list.
  - Traverse off the end and be at the front of the list.
- Requires a tail pointer instead of a headptr.

```
struct node
{
    int num;
    node *next;
};
```

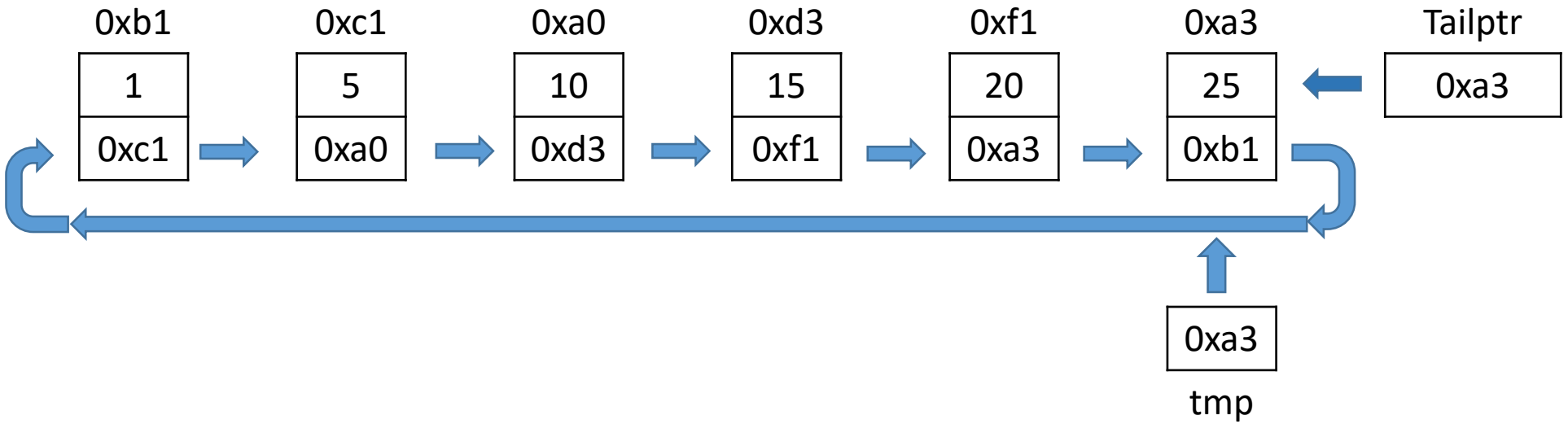
SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Traversal



SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

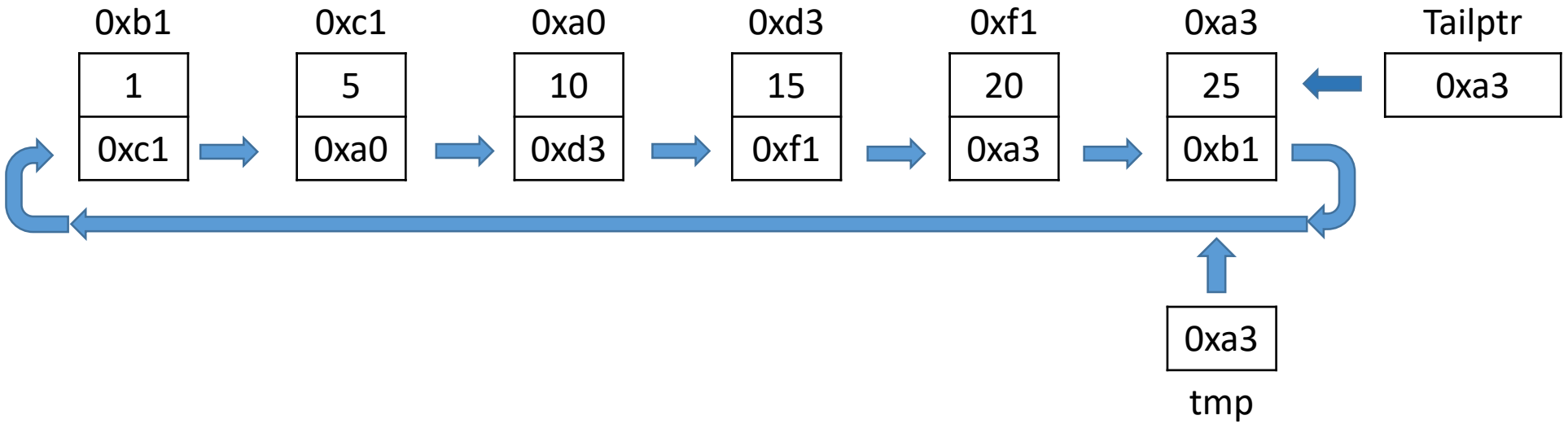
Traversal



Set temp pointer to the value of tail pointer

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

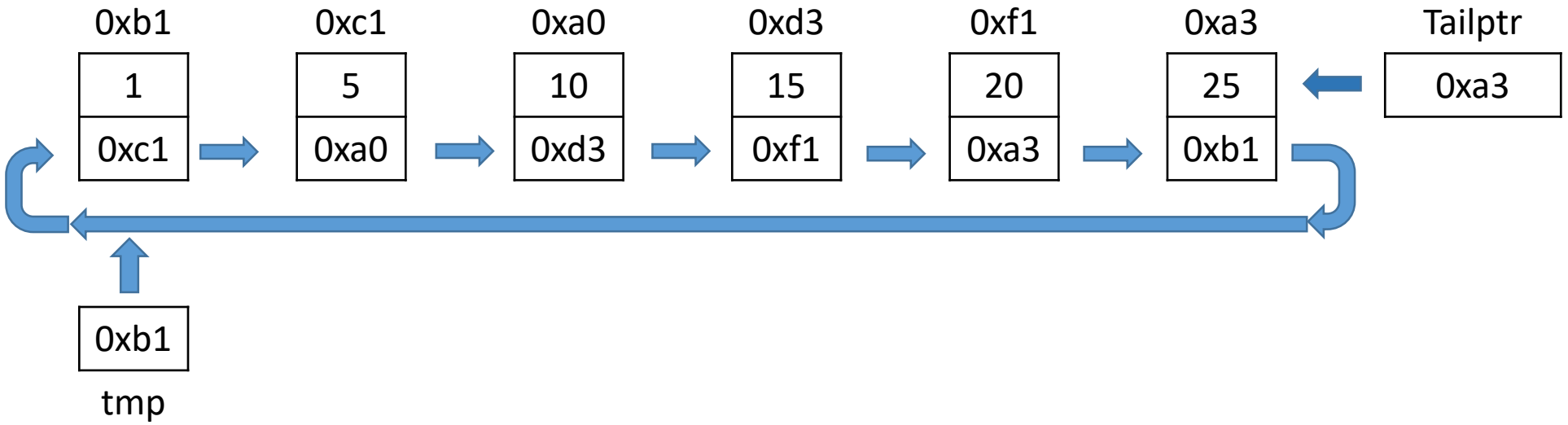
Traversal



Move temp pointer down, print that item and repeat if not the tail pointer

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Traversal

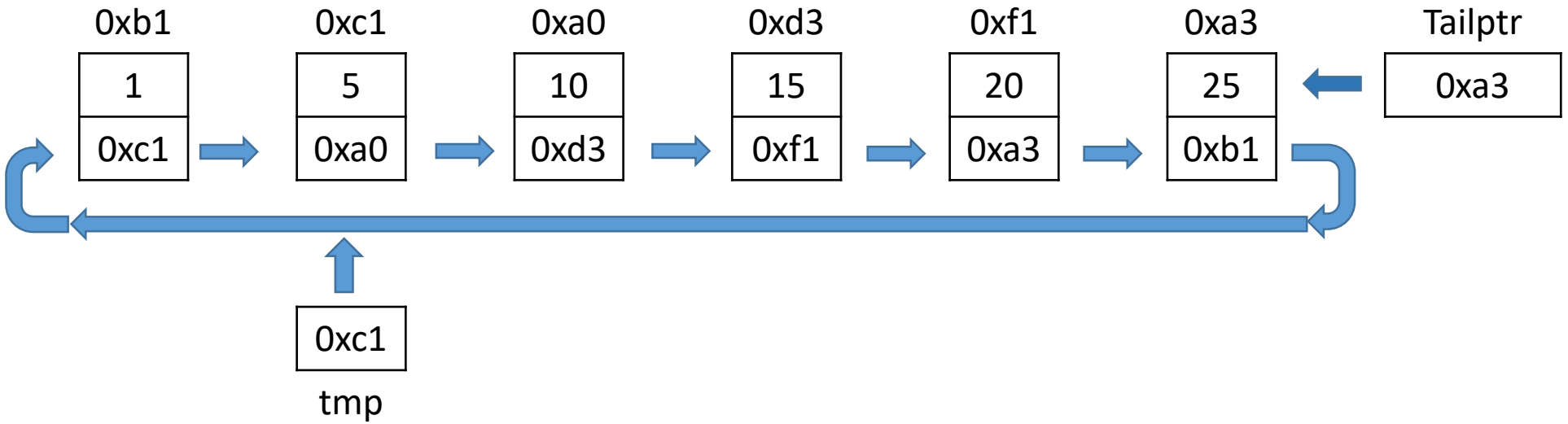


Output: 1

Move temp pointer down, print that item and repeat if not the tail pointer

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Traversal

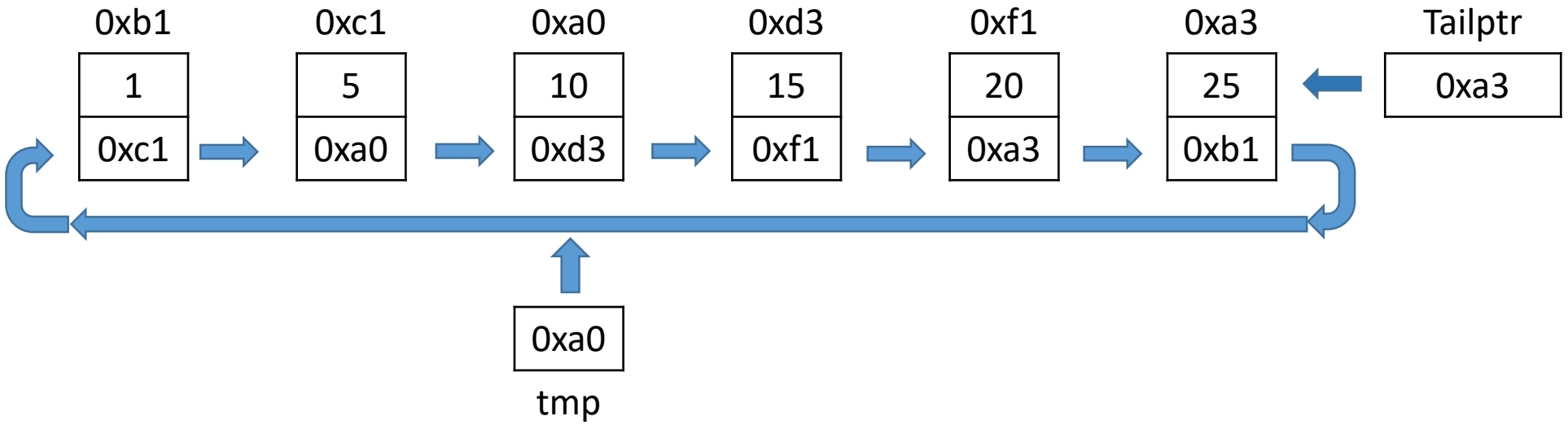


Output: 1 5

Move temp pointer down, print that item and repeat if not the tail pointer

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Traversal

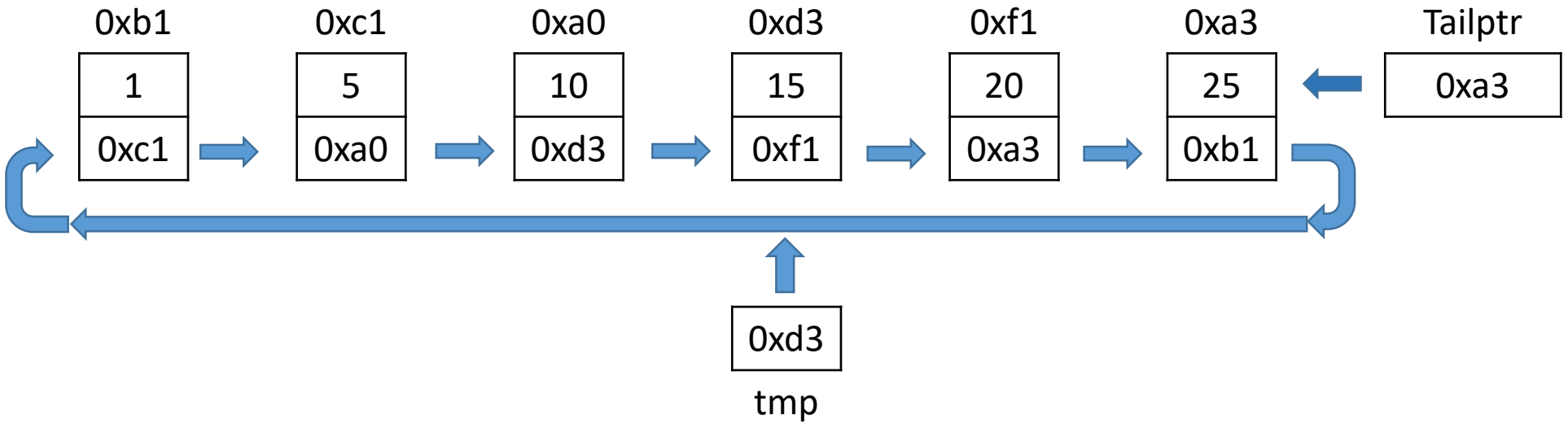


Output: 1 5 10

Move temp pointer down, print that item and repeat if not the tail pointer

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Traversal

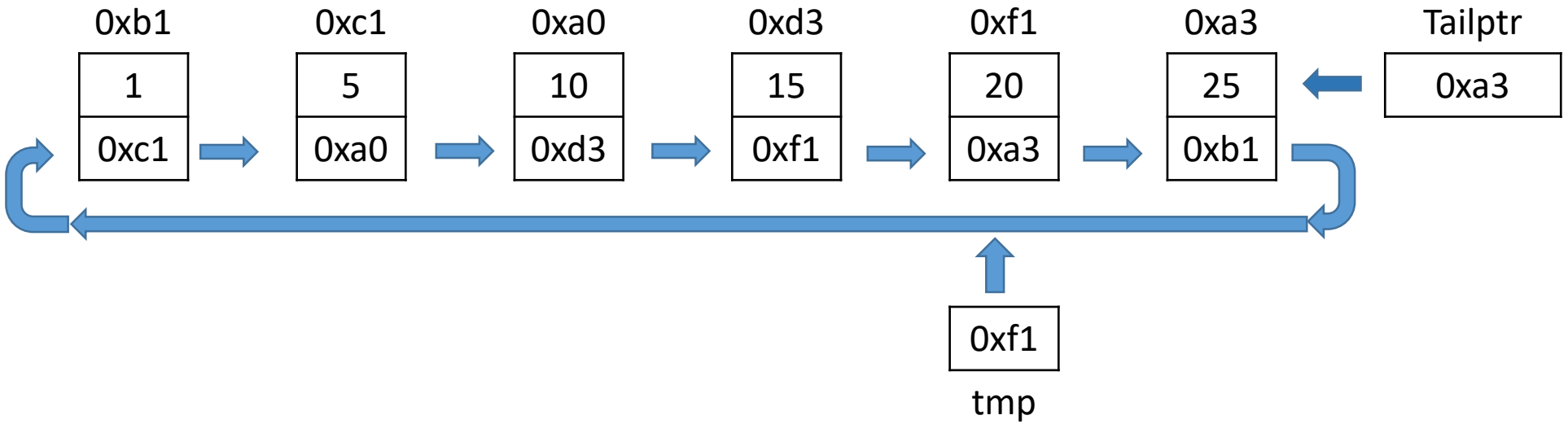


Output: 1 5 10 15

Move temp pointer down, print that item and repeat if not the tail pointer

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Traversal

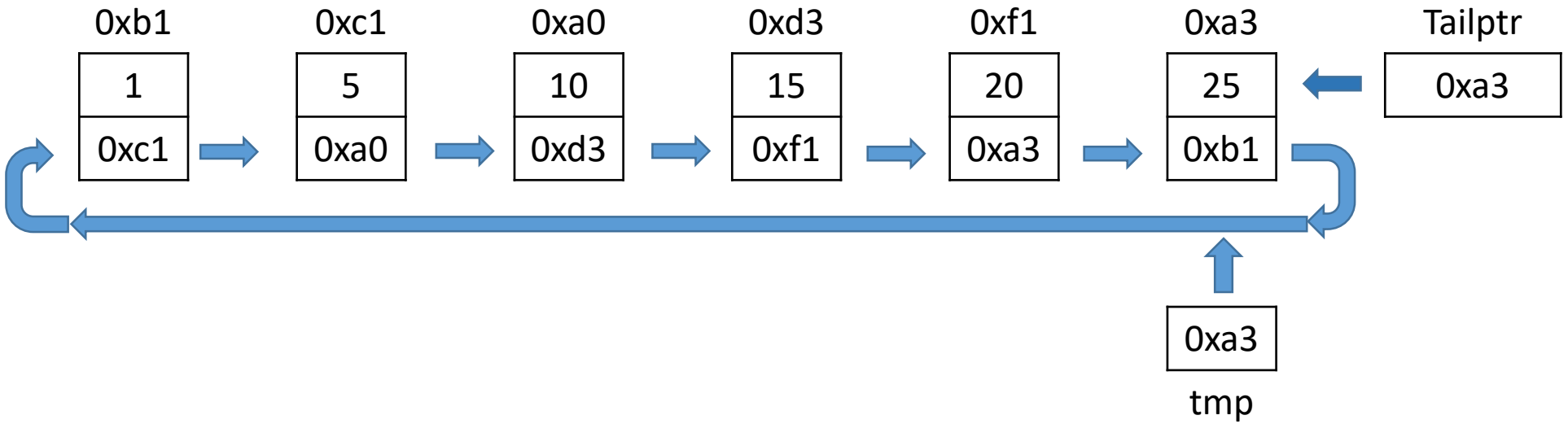


Output: 1 5 10 15 20

Move temp pointer down, print that item and repeat if not the tail pointer

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Traversal



Output: 1 5 10 15 20 25

Move temp pointer down, print that item and repeat if not the tail pointer, stops.

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

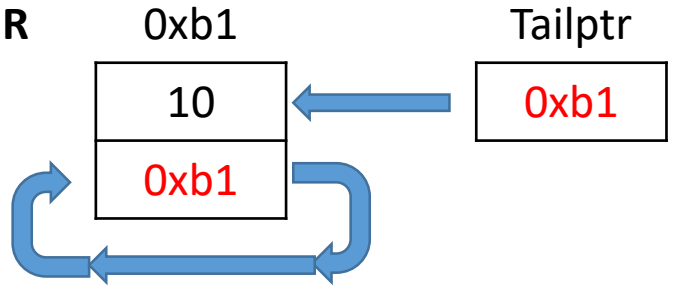
Insertion

1. Empty
2. Front
3. Middle
4. End

Insert - Empty (10)

BEFORE

AFTER

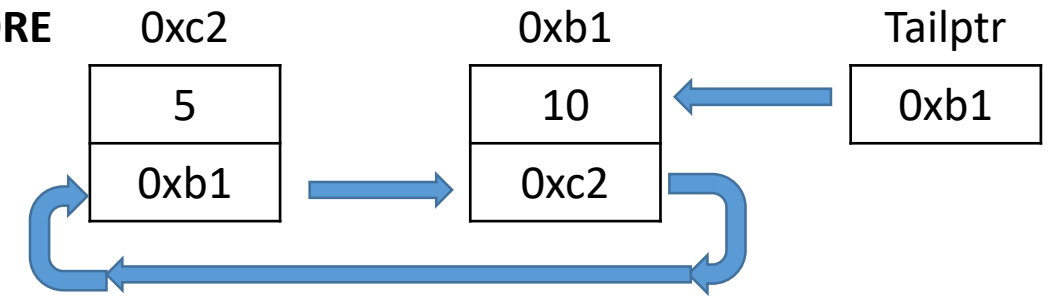


SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

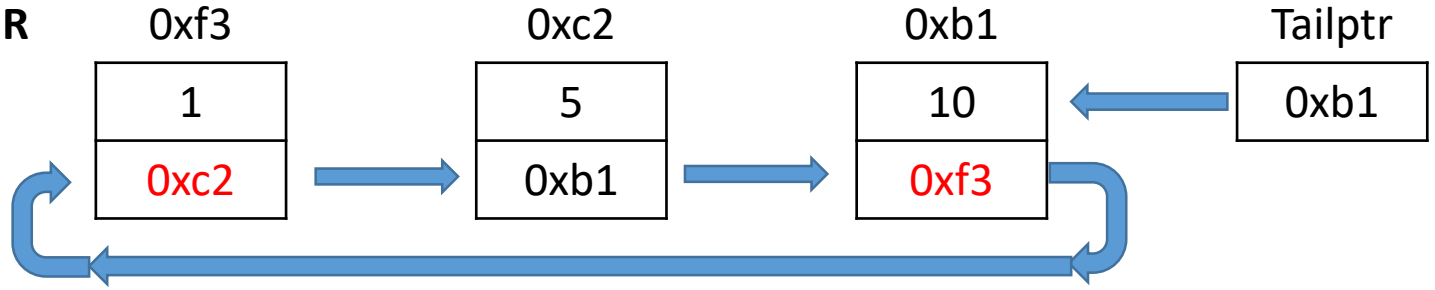
Insert - Front (1)

Assume 5 and 10 are in list for clarity

BEFORE

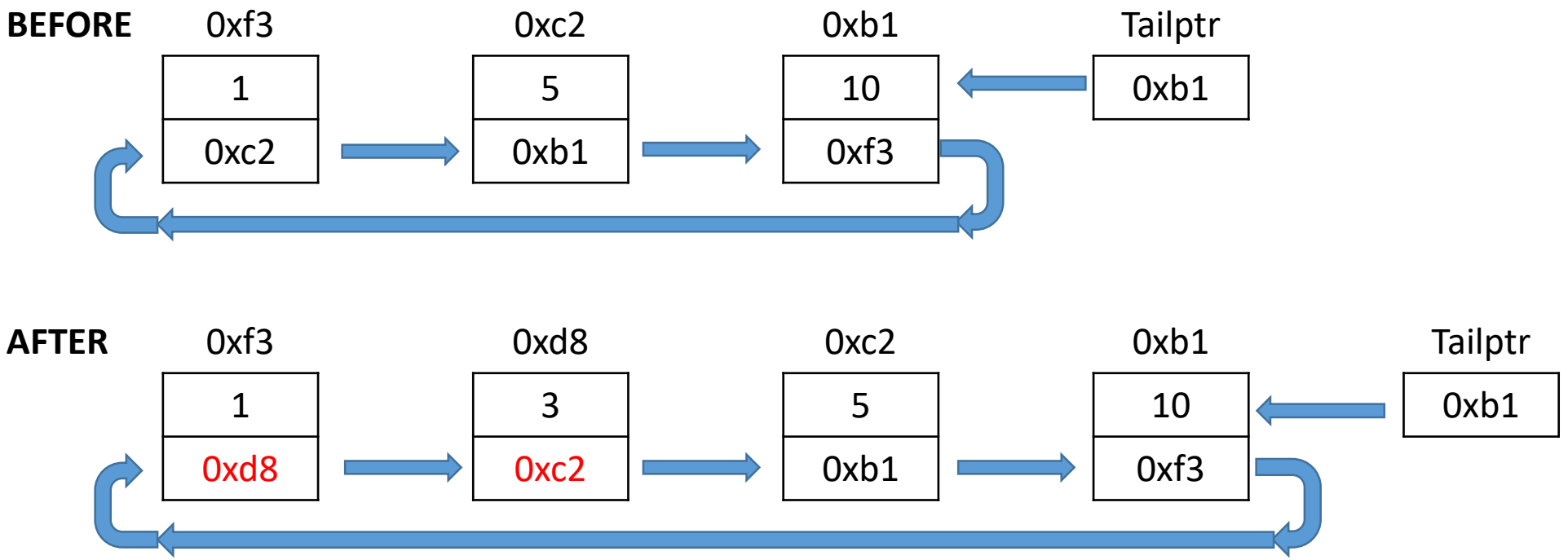


AFTER



SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Insert – Middle (3)

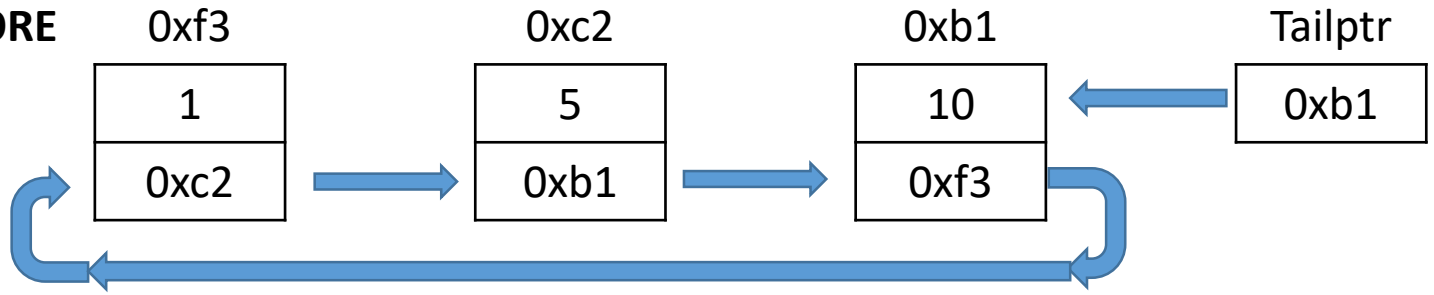


SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

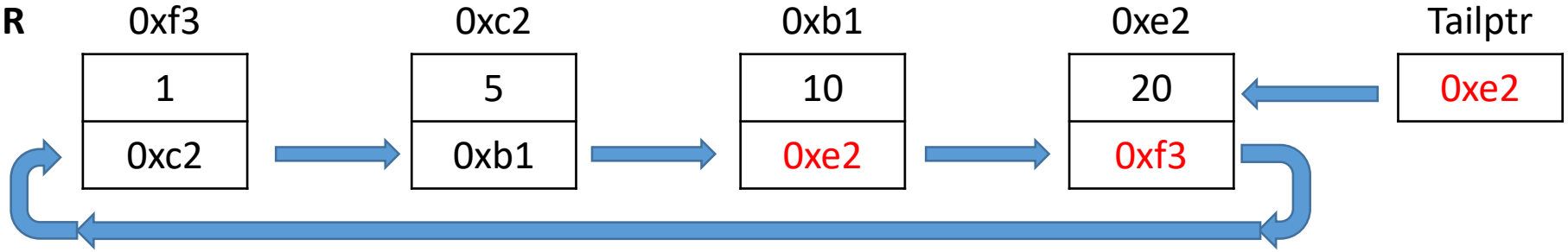
Insert – End (20)

Assume 1, 5, and 10 are in list for clarity

BEFORE



AFTER



SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

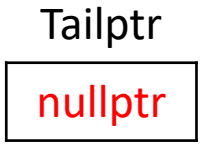
Removal

1. Empty
2. Front
3. Middle
4. End
5. Last node in list

Remove – Empty

BEFORE

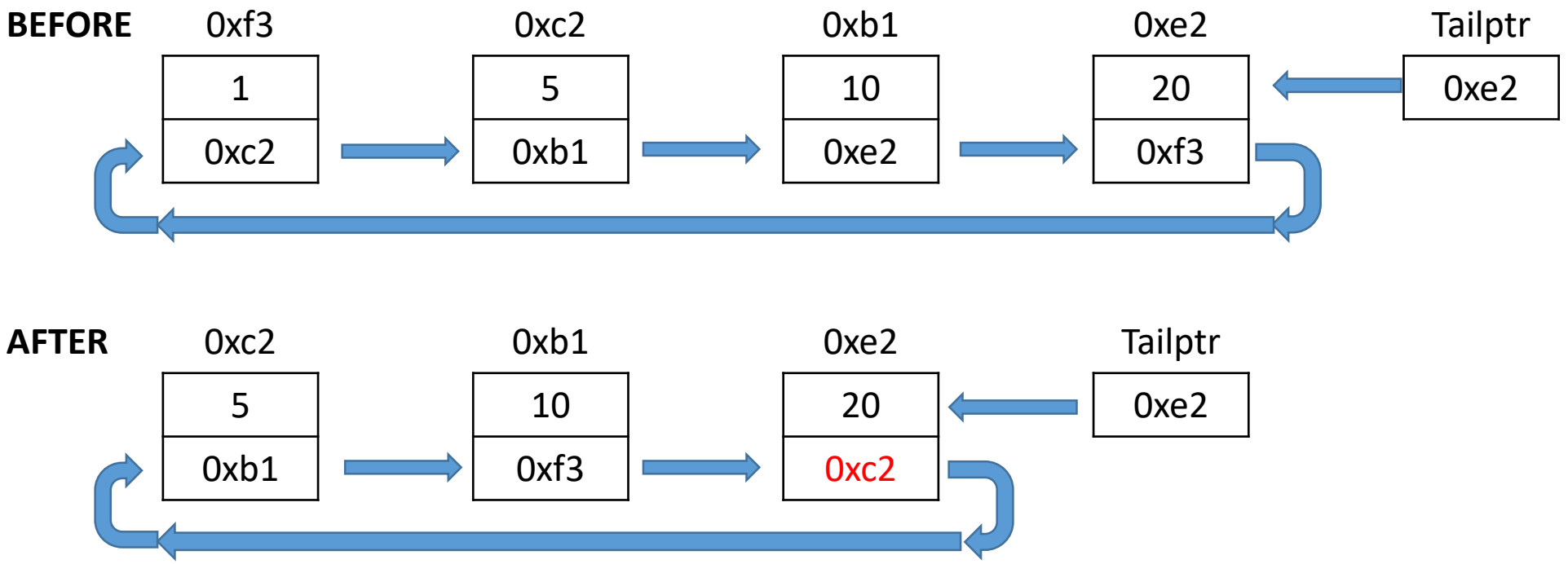
AFTER



Should return false

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

Remove – Front (1)

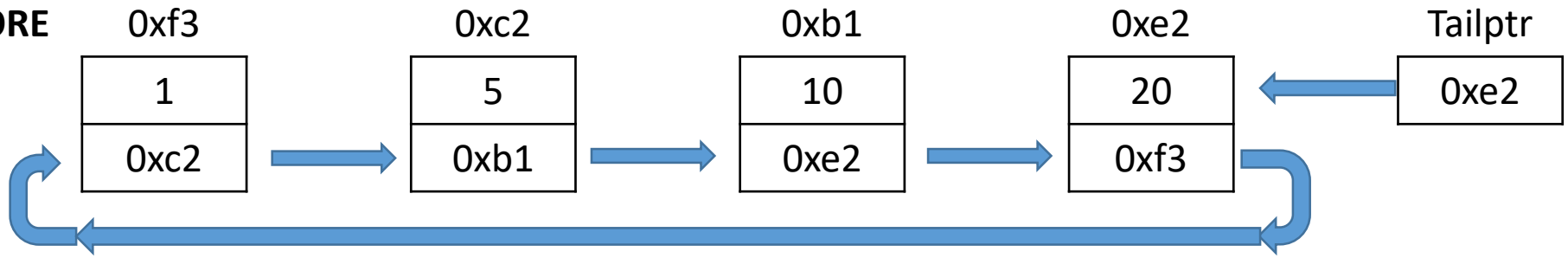


SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

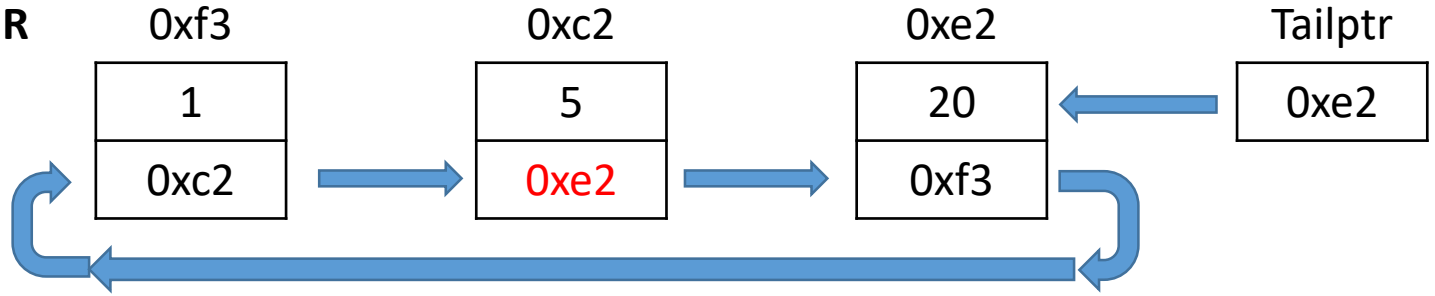
Remove – Middle (10)

Assume 1, 5, and 10 are in list for clarity

BEFORE



AFTER

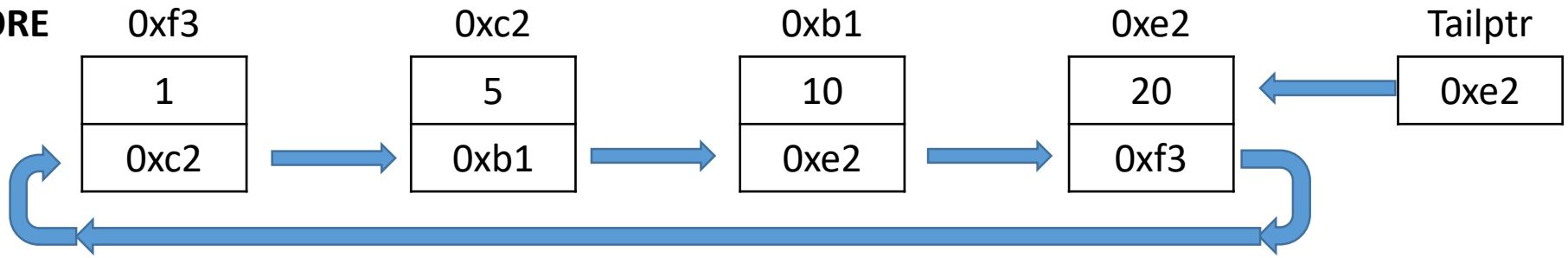


SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

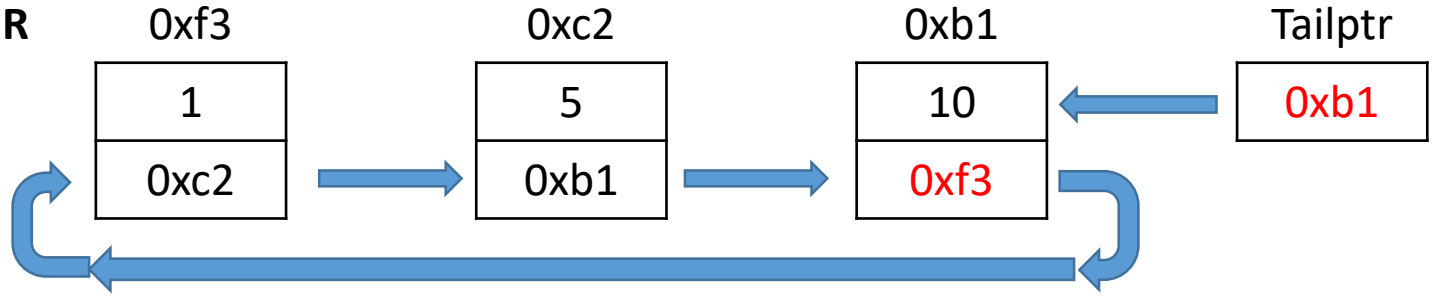
Remove – End (20)

Assume 1, 5, and 10 are in list for clarity

BEFORE



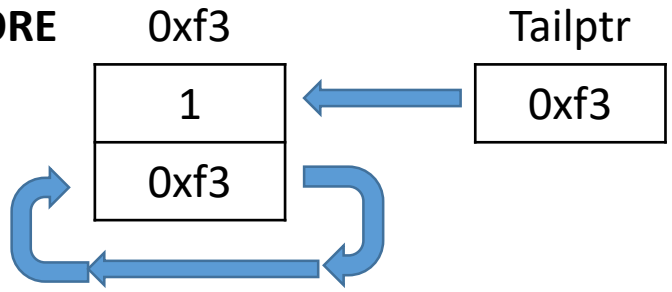
AFTER



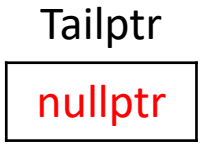
SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Remove – Last Node in List

BEFORE



AFTER



SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY