601.220 Intermediate Programming

Summer 2024, Meeting 12 (June 28)

Today's agenda

- Review of exercises 17 and 18
- Work on midterm project

Updates about midterm project

- Deadline for midterm project was be moved one day. New deadline is **Saturday June 29th at 11pm**.
- No further extensions will be provided for the project.
- Deadline for the indiviual contributions survey is still **Sunday, June 30rd at 11pm**. ⇝ Gradescope
- **Reminder**:Late days cannot be used for the midterm or final project.
- Only one team member needs to submit the code.. Everybody fills the individual contributions survey.

Reminders/Announcements

- Midterm exam: in class on **Wednesday, July 3th**, see full post in Piazza.
  - Exam details:
    - *Synchronous*, i.e., you *must* attend the Zoom meeting
    - You will work in a breakout room with your camera on
    - Access to internet resources, editor/compiler, etc. is allowed
    - Communication with or help from other people is prohibited
  - **NOT** Allowed during the exam:
    - The use of AI-based tools, such as ChatGPT, GitHub copilot, and GitHub copilot chat, is strictly prohibited. Visual studio code will not be allowed during the exam due to its tight integration with GitHub Copilot.
  - Study materials: previous year review session and practice exams..

Exercise 17 review

Node data type:

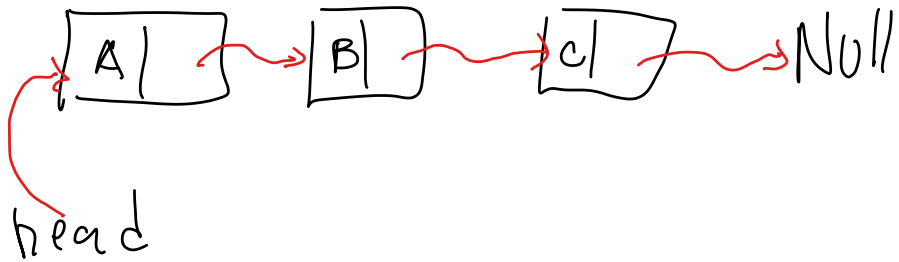
```
typedef struct Node_ {
  char data;
  struct Node_ *next;
} Node;
```

The `typedef` allows us to refer to the "`struct Node_`" type as just "`Node`".

Exercise 17 review (Length function)

```
// length function, while loop version
int length(const Node *n) {
  int count = 0;
  while (n != NULL) {
    count++;
    n = n->next;
  }
  return count;
}
```

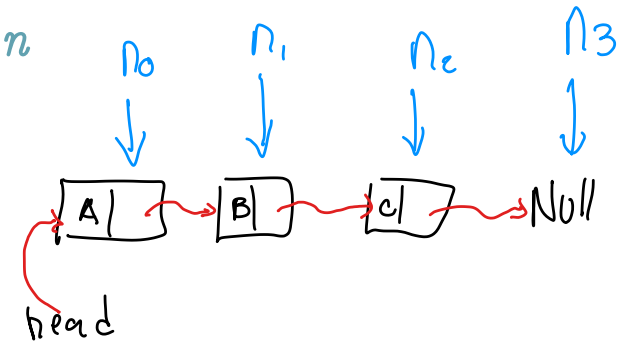
n->val = 'm';



Note: `const Node *n` means "`n` is a pointer to `const Node`". Function is saying that it won't modify the object that `n` points to.

Exercise 17 review (Length function recursive)

```
// length function, recursive version
int length(const Node *n) {
  if (n == NULL) {
    return 0;
  }
  return 1 + length(n->next);
}
```



A linked list can be considered as a *recursive* data structure.

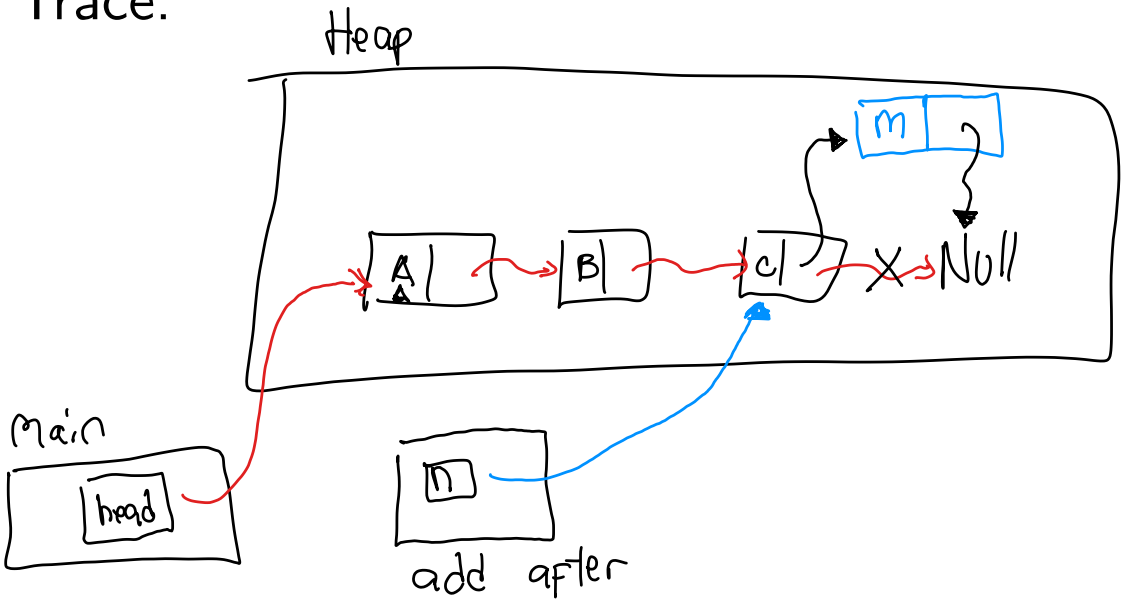
Assume `n` is a pointer to a linked list node. Cases:

1. `n` is NULL: the list is empty
2. `n` points to a node: nonempty list, `n->next` points to a smaller list (with one fewer nodes than the overall list)

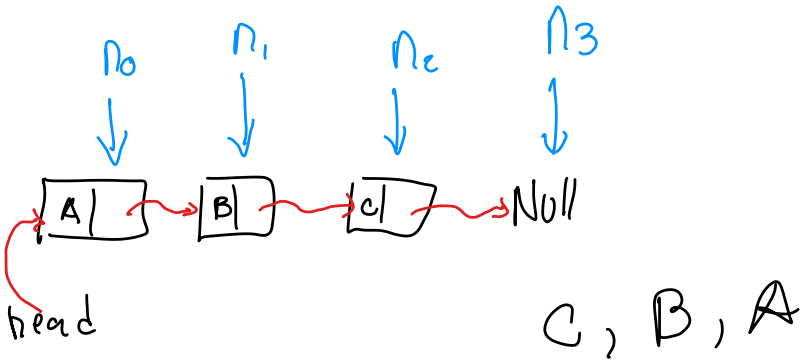
Exercise 17 review

```
void add_after(Node *n, char value) {
  const Node *node = malloc(sizeof(Node));
  node->data = value;
  node->next = n->next;
  n->next = node;
}
```

Trace:



Exercise 17 review




```
void reverse_print(const Struct Node *n) {
  // Pseudo code:
  // if (n is the empty list)
  //   do nothing, return
  // else
  //   print the rest of the list in reverse order
  //   print the value of the first element
}
```

Exercise 18 review

```
void remove_after(Node *node) {
  Node *removed = node->next;
  if (removed == NULL) { return '?'; }

  node->next = removed->next;
  char result = removed->data;
  free(removed);
  return result;
}
```

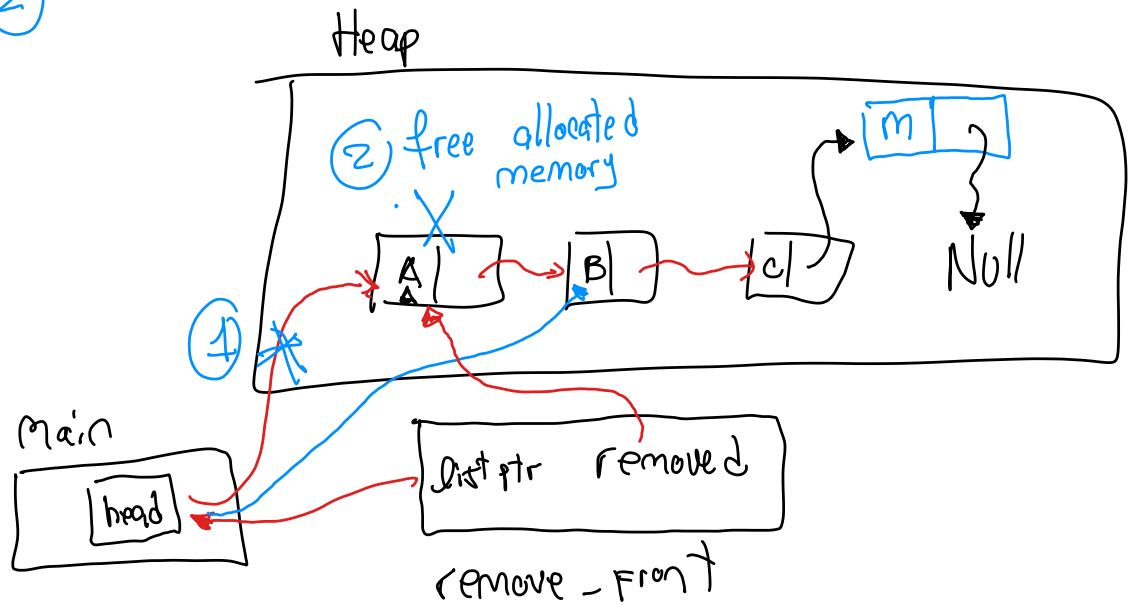
Trace:

Exercise 18 review

```
char remove_front(Node **list_ptr) {
  if (*list_ptr == NULL) { return '?'; }

  Node *removed = *list_ptr;
  *list_ptr = removed->next;
  char result = removed->data;
  free(removed);
  return result;
}
```

Trace:



Exercise 18 review

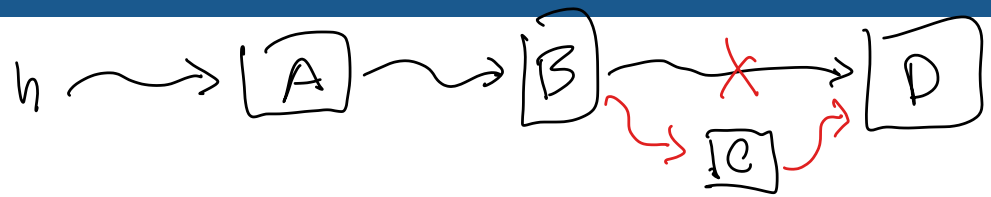
```
void remove_all(Node **list_ptr, char val) {
  if (*list_ptr == NULL) return; // reached end of list?

  if ((*list_ptr)->data == val) {
    // remove first element

  } else {
    // skip first element

  }
  remove_all(list_ptr, val); // remove remaining occurrences
}
```

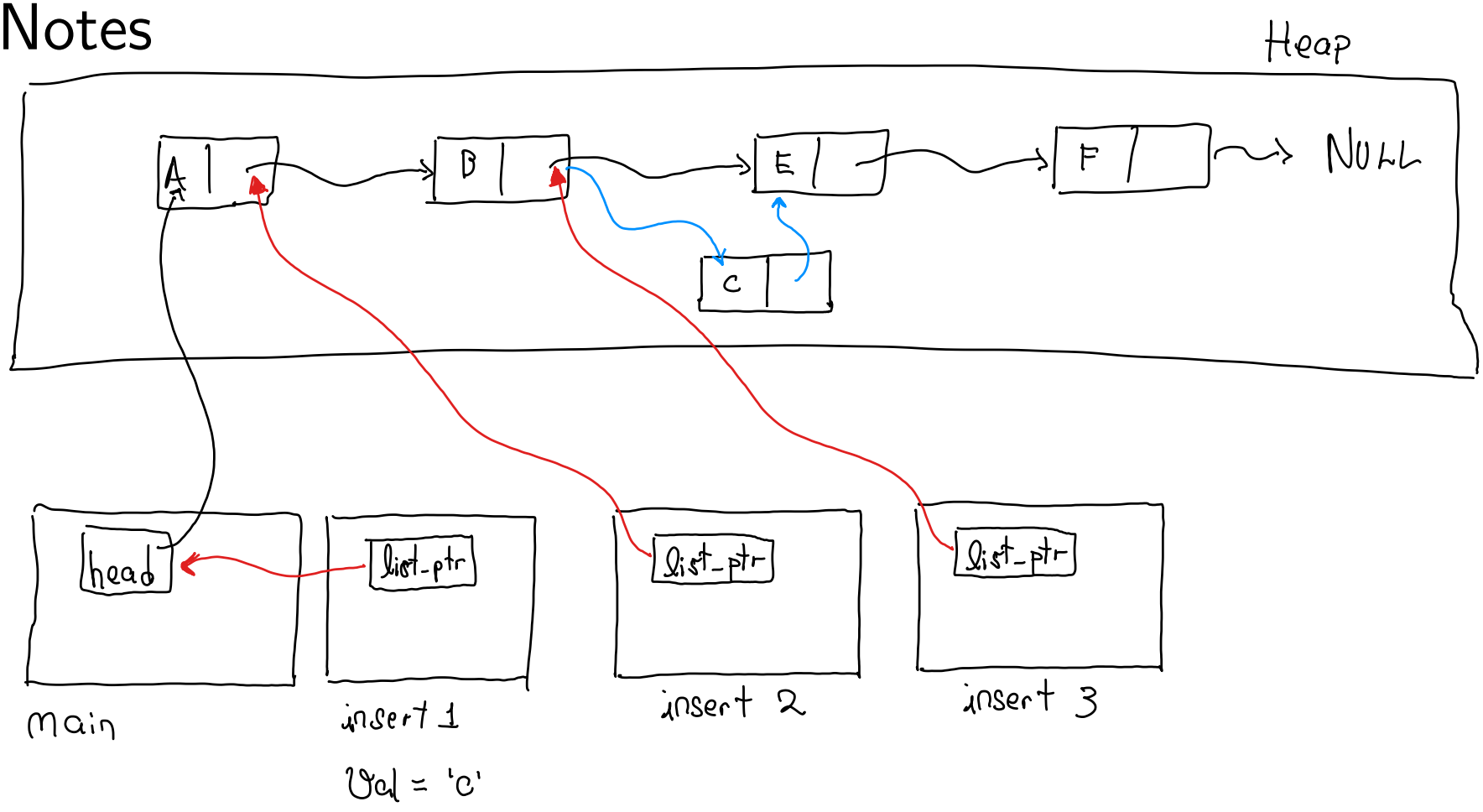
Exercise 18 review




```
Node *insert(Node **list_ptr, char val) {
  if (*list_ptr == NULL || val < (*list_ptr)->data) {
    add_front(list_ptr, val);
    return *list_ptr;
  } else {
    // recursion
    insert ( _____ , val );
    return *list_ptr;
  }
}
```

* I will be providing 5 extra points that can be used in any hw to whoever guess first the expression in the recursion block.
* deadline: by the end of the class.

Notes


Heap
A
B
E
F
NULL
C
head
list_ptr
list_ptr
list_ptr
main
insert 1
Val = 'c'
insert 2
insert 3

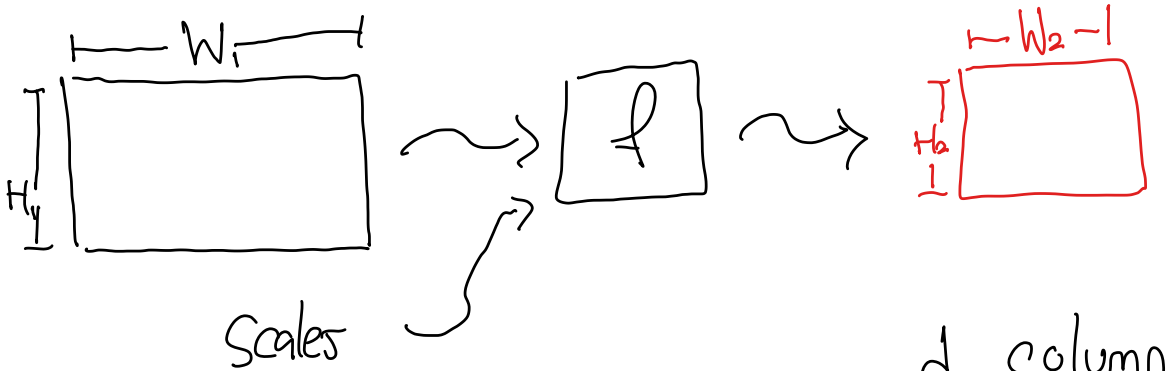
Work on midterm project!

- You can also ask questions about exercises and/or exam review material
- Breakout rooms 1–10 are "social"
- Use Slack to let us know if you have a question
  - This is preferred: the CAs have no way of seeing the Zoom "ask for help" feature

Notes Seam carving notes

$W_2 + d = W_1$

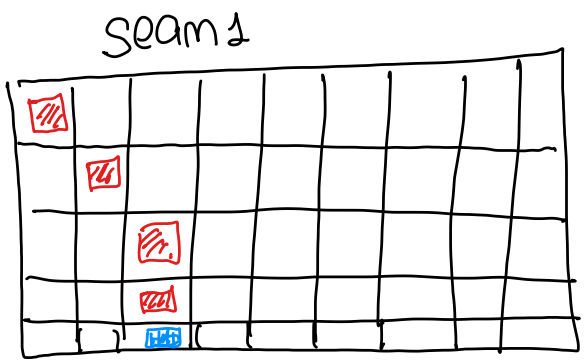
$d = \text{int}(W_1 * \text{scale})$



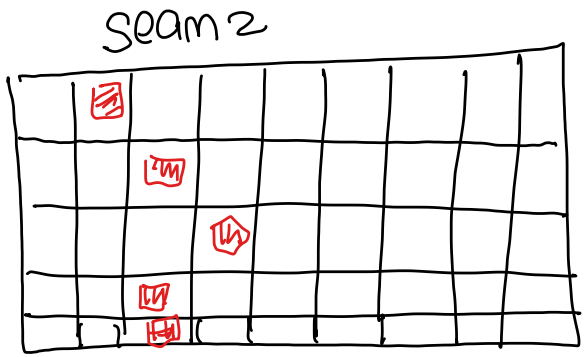
$W_2 \leq W_1$

$H_2 \leq H_1$

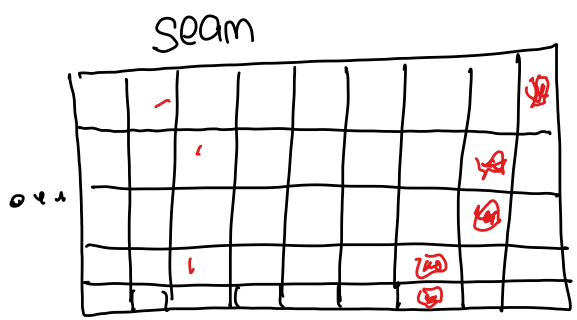
d columns need to be removed.



Sm 1



Sm 2



Sm 3

Notes

Notes

Notes