

# CS 162

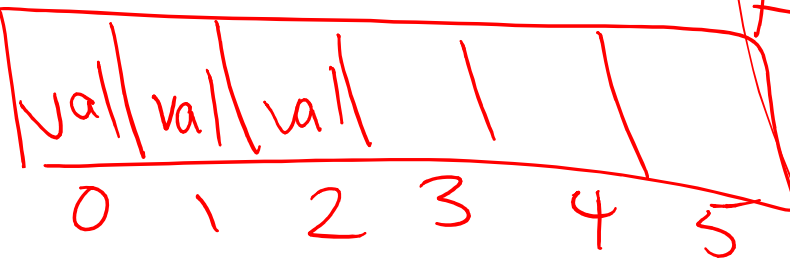
## Intro to CS II

Intro to C Programming

# Linked Lists

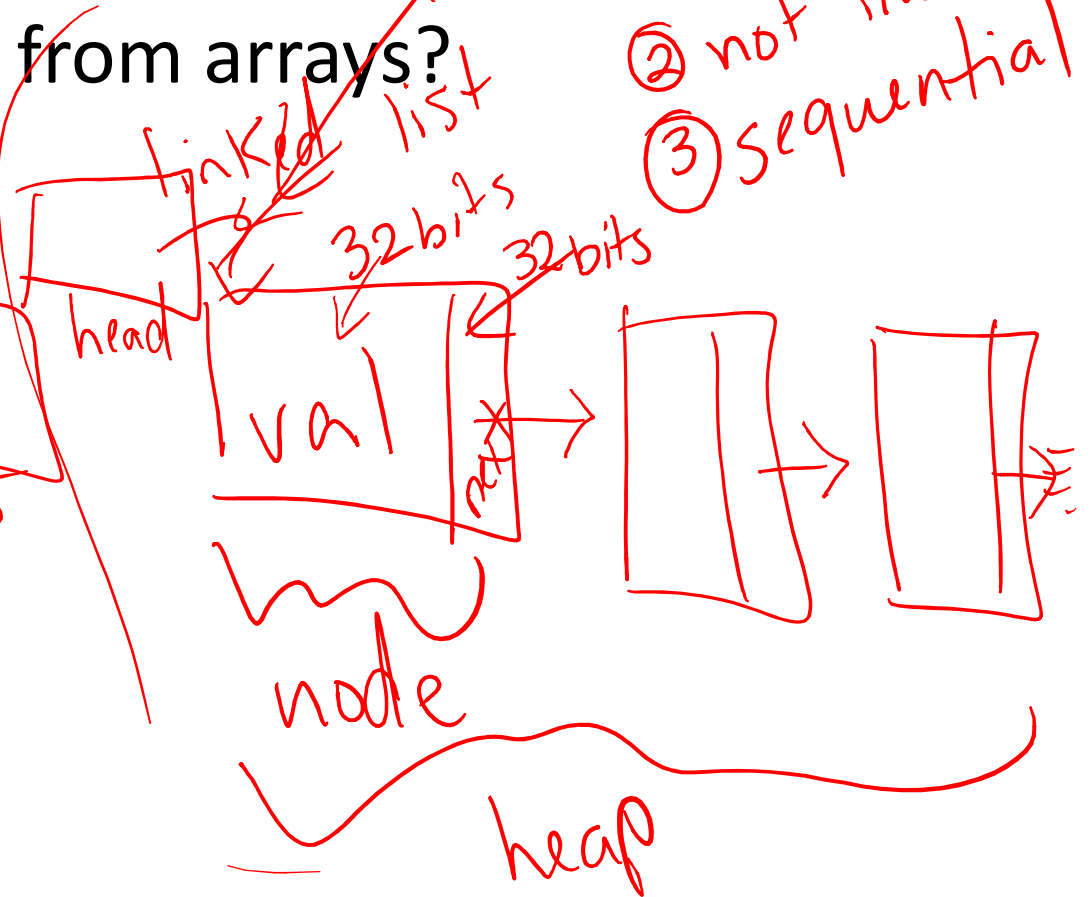
- What are they?
- How do they differ from arrays?

array



~~$O(1)$  constant time~~  
inserting a new value to the beginning

- ① not contiguous
- ② not indexed
- ③ sequential



```
access.engr.orst.edu - PuTTY
1 #include <stdio.h>
2 #include <stdlib.h> //NULL is here
3
4 struct node {
5     int val;
6     struct node *next;
7 };
8
9 int main(){
10     int num, i;
11     struct node *head=NULL;
12
13     //head = new node;
14     head=malloc(sizeof(struct node));
15
16     head->val=20;
17     head->next=NULL;
18
19     printf("1st node val: %d\n", head->val);
20     //delete head;
21     free(head);
22
23     printf("hello\n");

```

Handwritten notes and diagrams:

- Stack:** A box labeled "stack" with "head" and "0x100" written inside. A circled "1" is next to it.
- Heap:** A box labeled "heap" with "val" and "next" fields. The "val" field contains "20" and the "next" field contains "1". A circled "2" is next to it.
- Diagram:** A box labeled "0x2000" with "0x100" inside. An arrow points from "0x100" to the "next" field of the "heap" box.
- Text:** "ins ptr" is written above the "stack" box. "ins ptr = head" is written above the "heap" box.
- Text:** "head = malloc" is written below the "stack" box.
- Text:** "head->next = ins ptr" is written below the "heap" box.
- Text:** "0x164" is written below the "heap" box.
- Text:** "19,4" and "Top" are written at the bottom right.

```
1 #include <stdio.h>
2 #include <stdlib.h> //NULL is here
3
4 struct node {
5     int val;
6     struct node *next;
7 };
8
9 int main() {
10     int num, i;
11     struct node *head=NULL, *insptr=NULL;
12
13     //head = new node;
14     head=malloc(sizeof(struct node));
15
16     head->val=20;
17     head->next=NULL;
18
19     insptr=head;
20     head=malloc(sizeof(struct node));
21     head->next=insptr; //link to old node
22     head->val=10;
23     insptr=NULL;
24
25     printf("1st node val: %d\n", head->val);
26     printf("2nd node val: %d\n", head->next->val);
27     //delete head;
28     free(head->next);
29     free(head);
```

# Fun Friday

- Get into groups of 4-5.
- Write the pseudocode for pushing a node with a new value onto the **front** of the list.
- Write the pseudocode for pushing a node with a new value onto the **back** of the list.