Linked Lists

Nalini Vasudevan
Columbia University

struct data type

```
struct sname
{
   int a;
   char b;
};
```

struct data type

```
struct sname
{
   int a;
   char b;
};

main ()
{
   struct sname s;
   s.a = 5;
   s.b = 'x';
}
```

struct data type

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

Linked list

```
main()
{
    struct node *head, *middle, *tail;
    head = malloc (sizeof(struct node));
    middle = malloc (sizeof (struct node));
    tail = malloc (sizeof (struct node));
    head → val = 1;
    middle → val = 2;
    tail → val = 3;
    head → next = middle;
    middle → next = tail;
    tail → next = NULL;
}
```

Printing the linked list

```
struct node *t;
t = head;
printf (“%d”, t->val);
t = t ->  next;
printf (“%d”, t->val);
t = t → next;
printf (“%d”, t->val);
```

Printing the linked list

```
struct node *t;
t = head;
while (t != NULL)
{
   printf (“%d”, t->val);
   t = t ->  next;
}
```

Creating a linked list

```
head = malloc (sizeof(struct node));
head → val = 1;
prev = head;
for (i = 2; i <= n; i++)
 {
   cur = malloc (sizeof (struct node));
   cur → val = i;
   prev → next = cur;
   prev = cur;
}
prev → next = NULL;
```

Deleting a Node

```
prev = head;
cur = head → next
while (cur != NULL)
{
    if (cur->val == key)
    {
        prev → next = cur → next;
        free(cur);
        break;
    }
    prev = cur;
    cur = cur → next;
}
```

Deleting a Node

```
if (head-> val == key)
{
   temp = head;
   head = head → next;
   free(temp);
}
else {
   ..
}
```

More operations

- Insert in the beginning of a list
- Insert in the middle of a list
- Insert in sorted order
- Concatenate two lists