CS 211 - 2/23/18

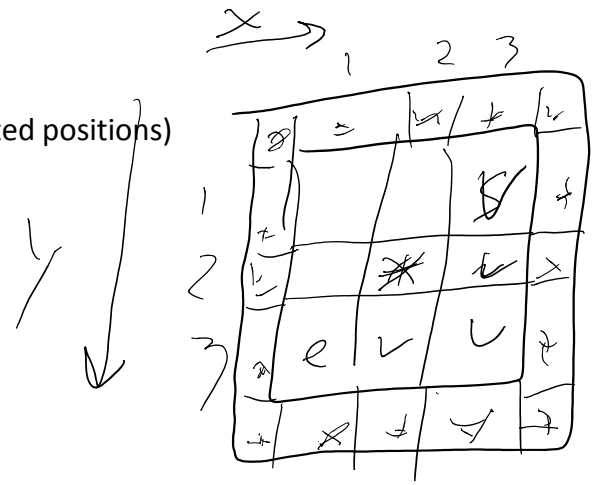
Friday, February 23, 2018 9:52 AM

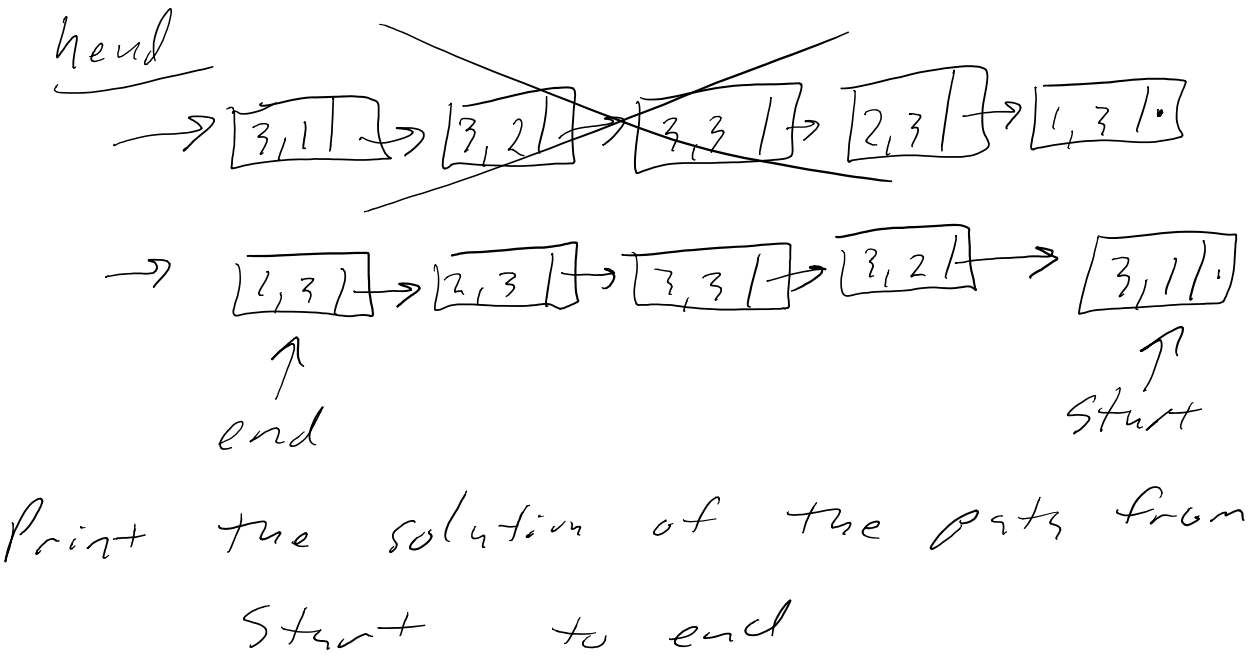
Next Wednesday: Engineering Career Center's Tech Career Fair 12noon - 4pm

For Project 3, we need two data stores

1. Maze info (blocked positions, visited/unvisited positions)
   dynamic 2-D character array

2. Stack containing the path
   linked list





Print the solution of the path from start to end

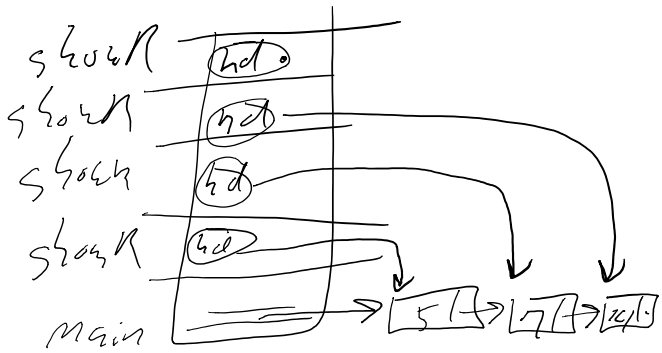
```
Stack st1, st2;

…
/* st1 contain path from end to start */

/* flip arouind the path in st2 */
st2 = int();

while ( !isEmpty(st1) )
  {
```

```
    push ( st2 , top ( st1 ) ) ;
    pop ( st1 ) ;
   }
/* now st2 has a linked list from Start to End */
```

Investigate Recursive linked list code

```
/* non recursive version to displau all the values in the linked list */
void show (node* hd)
{
 /* loop until the PBV parameter is NULL */
 while (hd != NULL)
  {
   printf ("%d, ", hd->elem);   /* access the value at the node */
   hd = hd->next;              /* move to the next node */
  }
 printf ("\n");
}
```

```
/* rewrite show() recursively */
/*   Base case:   hd is NULL  => just print the \n and return */
/*  Recursive case:   hd != NULL  =>  print value at hd->elem
                                       make a recursive call with hd->next
void showR( node* hd)
{
 if ( hd == NULL )
   {
     printf ("\n");
     return;
   }
 else
   {
    printf ("%d, " hd->elem);
    showR ( hd->next);
    return;
   }
}
```




```
/*  showRR() Recursively in Reverse */
/*   Base case:   hd is NULL  =>  return */
/*  Recursive case:   hd != NULL  =>   make a recursive call with hd->next
                                       print value at hd->elem    */

void showRR( node* hd)
{
 if ( hd != NULL )
```

```
void showRR( node* hd)
{
 if ( hd != NULL )
    {
     showRR ( hd->next);
     printf ("%d, " hd->elem);
    }
}
```

Output: 14, 7, 5,

```
typedef struct nodeStruct
{
 int elem;
 struct nodeStruct*  next;
} node;
```