Fundamentals of Computing: Lecture 9

Piyush P Kurur
Office no: 224
Dept. of Comp. Sci. and Engg.
IIT Kanpur

August 17, 2009

Summary of last class

Summary of last class

- We looked at the tower of hanoi problem.

Summary of last class

- We looked at the tower of hanoi problem.
- We gave a recursive solution to that problem.

Summary of last class

- We looked at the tower of hanoi problem.
- We gave a recursive solution to that problem.

Another example

Given an integer n print the sequence
$n, n - 1, \ldots, 1, 0, 1, \ldots, n - 1, n$

Using iteration

```c
# include<stdio.h>
void downUp(int n)
{
  for(int i = n; i > 0; i--)
  {
    printf("%d, ", i);
  }
  printf("0");
  for(int i = 1; i <= n; i ++)
  {
    printf(", %d", i);
  }
}
```

Let S_n denote the sequence $n, \ldots, 0, \ldots, n$.

Recursive definition

$$
\begin{aligned}
S_0 &= 0 \\
S_n &= n, S_{n-1}, n
\end{aligned}
$$

Let S_n denote the sequence $n, \ldots, 0, \ldots, n$.

Recursive definition

$$
\begin{aligned}
S_0 &= 0 \\
S_n &= n, S_{n-1}, n
\end{aligned}
$$

```c
void downUpRec(int n)
{
  if(n <= 0)
  {
    printf("0");
    return;
  }
  printf("%d, ", n);
  downUpRec(n-1);
  printf(", %d", n);
}
```

The `main` function

```
void downUpRec(int);
void downUp(int );

int main()
{
  int n;
  printf("enter the number: ");
  scanf("%d",&n);
  printf("Iterative:\t");
  downUp(n);
  printf("\nRecursive:\t");
  downUpRec(n);
  printf("\n");
}
```

Separate compilation

- Create a file `downUp.c` that contains the definition of the function `void downUp(int)`
- Create a file `downUpRec.c` that contains the definition of the function `void downUpRec(int)`
- Create a file `main.c`

Compiling

```
$ gcc -std=c99 main.c downUp.c downUpRec.c
$ gcc -std=c99 -c downUp.c
$ gcc -std=c99 -c downUpRec.c
$ gcc -std=c99 main.c downUp.o downUpRec.o
```