Sunbeam Institute of Information Technology
Pune and Karad

Module – Data Structures and Algorithms

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Recursion

- Function calling itself is called as recursive function.
- To write recursive function consider
  - Explain process/formula in terms of itself
  - Decide the end/terminating condition
- Examples:
  - $n! = n * (n-1)!$ $0! = 1$
  - $x^y = X * x^{y-1}$ $x^0 = 1$
  - $T_n = T_{n-1} + T_{n-2}$ $T_1 = T_2 = 1$
  - factors(n) = 1st prime factor of n * factors(n)

```
int fact(int n) {
  int r;
  if(n==0)
    return 1;
  r = n * fact(n-1);
  return r;
}

int main() {
   int res;
   res = fact(5);
   printf("%d", res);
   return 0;
}
```

Recursion execution

```
int fact(int n) {    int fact(int n) {    int fact(int n) {    int fact(int n) {    int fact(int n) {    int fact(int n) {
  int r;               int r;               int r;               int r;               int r;               int r;
  if(n==0)             if(n==0)             if(n==0)             if(n==0)             if(n==0)             if(n==0)
    return 1;            return 1;            return 1;            return 1;            return 1;            return 1;
  r = n * fact(n-1);   r = n * fact(n-1);   r = n * fact(n-1);   r = n * fact(n-1);   r = n * fact(n-1);   r = n * fact(n-1);
  return r;            return r;            return r;            return r;            return r;            return r;
}                    }                    }                    }                    }                    }
```

```
int main() {
   int res;
   res = fact(5);
   printf("%d", res);
   return 0;
}
```

5! = 5 * 4!

4! = 4 * 3!

3! = 3 * 2!

2! = 2 * 1!

1! = 1 * 0!

0! = 1

Sorting Algorithm : Selection Sort

Algorithm:

- Find the minimum element in an array A[i -> n-1] and place it at beginning
  - where n – size of array and i – 0, 1, 2, …n-2
- Repeat the above procedure n – 1 times where n is size of array

- Select ith element (i = 0 -> n-1)
  - Compare with all elements other than ith
    - if(A[i] > A[other])
      - Swap both elements

arr	44	11	55	22	66	33
	0	1	2	3	4	5

Sorting Algorithm : Bubble Sort

Algorithm:

- Find the maximum element from two consecutive elements of an array A[i -> n-i-1] and place it at second location
  - where n – size of array and i – 0, 1, 2, …n-2
- Repeat the above procedure n – 1 times where n is size of array

- Repeat for n-1 times
  - Compare two consecutive elements
    - If left element > right element
      - Swap both elements

arr	**33**	**22**	**66**	**55**	**44**	**11**
	0	1	2	3	4	5

Sorting Algorithm : Insertion Sort

Algorithm:

- Repeat from 1 to n-1
  - Select ith element in the array
    - Compare ith element with all its left neighbours
      - Insert at appropriate position

arr	**55**	**44**	**22**	**66**	**11**	**33**
	0	1	2	3	4	5

Thank you!

Devendra Dhande

<devendra.dhande@sunbeaminfo.com>