Fundamentals of Computing: Lecture 19

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

September 11, 2009

Sorting

Given a list of numbers arrange them in ascending order.
We have already seen a sorting algorithm.
Let us first write a function that swaps the *i*-th and *j*-th element of an array.

```
void swap(int *a, int i, int j)
{
  int temp;
  temp = a[i];
  a[i] = a[j];
  a[j] = temp;
}
```

```
for(int i = 0; i < n - 1; i ++)
{
  j = i + 1;
  while( j > 0  && a[j] < a[j-1] )
  {
    swap(a,j,j-1);
    j--;
  }
}
```

How many comparisons does it take ?

- In the best case.
- In the worst case.

Quick sort

- Choose a pivot element x
- Divide the rest of elements into two groups A and B such that
  - A consists of all elements less than x.
  - B consists of all elements greater than or equal to x.
- Sort A and B recursively and then the arange them in the order A, x, B.

```
void qsort(int *a , int start, int end)
{
  if( start >= end) return
  pivot = start;
  /* partition */
  l = start + 1;
  m = end
  do {
    while( l < m && a[l] < a[pivot] ) l ++;
    while( m > l && a[m] > a[pivot] ) m --;
    if( m > l) swap(a, l, m);
  }while(l < m);

  if( a[l] < a[pivot])
  {
    swap(a, l, pivot);
    qsort(a,start, l);
    qsort(a,m+1,end);
  }
```

How many operations?