Examples

CS10001: Programming & Data Structures

Pallab Dasgupta
Professor, Dept. of Computer Sc. & Engg.,
Indian Institute of Technology Kharagpur

Merge-Sort

```
void mergesort ( int a[ ], int lo, int hi )
{
        int m;
        if (lo<hi) {
                m=(lo+hi)/2;
                mergesort(a, lo, m);
                mergesort(a, m+1, hi);
                merge(a, lo, m, hi);
        }
}
```

Function Merge

```
void merge ( int a[ ], int lo, int m, int hi )
{
        int i, j, k;

        // copy both halves of a to auxiliary array b
        for (i=lo; i<=hi; i++) b[i]=a[i];

        i=lo; j=m+1; k=lo;
        // copy back next-greatest element at each time
        while (i<=m && j<=hi)
                if (b[i]<=b[j]) a[k++]=b[i++];
                else a[k++]=b[j++];

        // copy back remaining elements of first half (if any)
        while (i<=m) a[k++]=b[i++];
}
```

Recursive Permutation Generator

```
void perm (char list[ ], int i, int n)
{
        int j, tmp;
        if (i == n) {
                for (j=0; j<=n; j++) printf("%c", list[ j ]);
                printf("\n");
        }
        else {
                for (j=i; j <= n; j++) {
                        SWAP(list[ i ], list[ j ], tmp);
                        perm(list, i+1, n);
                        SWAP(list[ i ], list[ j ], tmp);
                }
        }
}
```

```
#define SWAP(x, y, t) ((t) = (x), (x) = (y), (y) = (t))
```

Transitive Closure

```
Transclosure ( int adjmat[ ][max], int path[ ][max] )
{
        for (i = 0; i < max; i++)
                for (j = 0; j < max; j++)
                        path[i][j] = adjmat[i][j];

        for (k = 0; k < max; k++)
           for (i = 0; i < max; i++)
              for (j = 0; j < max; j++)
                 if ((path[i][k] == 1)&&(path[k][j] == 1)) path[i][j] = 1;
}
```