REPORT FOR ASSIGNMENT 12

Shashwat Gupta — 14IE10028
12 April 2016

Indirect sorting

Searching for unique element : $O(n^2)$

```
int is_unique(int ar[], const int n)
{
    int i, j;
    for(i=0; i<n; ++i)
    {
        for (j = i + 1; j<n; ++j)
        {
            if (ar[i] == ar[j])
                return ar[i];
        }
    }
    return -INF;
}
```

Creating graph out of comparisions : $O(n^2)$

```
int srt(int ar[], const int n)
{
    int i, j;
    int ret = -INF;
    for(i=0; i<n; ++i)
    {
        for (j = i + 1; j<n; ++j)
        {
            if (ar[i] == ar[j])
                ret = ar[i];
            if (ar[i] > ar[j])
                addEdge(ar[i], ar[j]);
            else
                addEdge(ar[j], ar[i]);
        }
        valid[ar[i]] = 1;
    }
    return ret;
}
```

Topological Sort

A topological sort or topological ordering of a directed graph is a linear ordering of its vertices such that for every directed edge uv from vertex u to vertex v, u comes before v in the ordering.

Time = O(V + E)

```
void topsort()
```

```
{
    int i;
    for(i=0; i<100; ++i)
    {
        if (valid[i])
        _visit(i);
    }
}
```

Post numbering :

```
void _visit(int i)
{
    if (vis[i] == temp)
        return;
    if (vis[i] == 1)
        return;
    vis[i] = temp;
    ++num;
    int j;
    for(j=0; j<100; ++j)
    {
        if (gph.ar[i][j] == 1)
            _visit(j);
    }
    postnumb[i] = num;
    postnumbed[nnn++] = i;
    printf("%d ", i);
    vis[i] = 1;
}
```

Command line args :

```
  if (argc == 1)
    {
        printf("Invalid num args\nCommand line input was expected\
    nRunning with default number\n");
        putchar('\n');
        int i;
        for(i=0; i<n; ++i)
            printf("%d ", arr[i]);
        putchar('\n');
    }
    else if (argc == 2)
    {
        n = atoi(argv[1]);
        printf("In : ");
        gen(arr, n);
        putchar('\n');;
    }
    else
    {
        n = atoi(argv[1]);
        int i;
        printf("In : ");
        for(i=0; i<n; ++i)
            printf("%d ", arr[i] = atoi(argv[i + 2]));
```

```
24          putchar('\n');;
25      }
```