Competitive Programming notes

Druhan Shah (ShockWave)

January 14, 2022

Contents

1 Sorting **1**
1.1 Merge Sort (with Inversion count) 1

2 Graphs **2**
2.1 Depth First Traversal (base) 2
2.2 Breadth First Traversal (base) 2

3 Flows **2**

1 Sorting

1.1 Merge Sort (with Inversion count)

```
ll mergeSort(int arr[], int array_size);
ll _mergeSort(int arr[], int temp[], int left, int right);
ll merge(int arr[], int temp[], int left, int mid, int right);

ll mergeSort(int arr[], int array_size) {
    int temp[array_size];
    return _mergeSort(arr, temp, 0, array_size - 1);
}
ll _mergeSort(int arr[], int temp[], int left, int right) {
    ll mid, inv_count = 0;
    if (right > left) {
        mid = (right + left) / 2;
        inv_count += _mergeSort(arr, temp, left, mid);
        inv_count += _mergeSort(arr, temp, mid + 1, right);
        inv_count += merge(arr, temp, left, mid + 1, right);
    }
    return inv_count;
}

ll merge(int arr[], int temp[], int left, int mid, int right) {
    int i, j, k;
    ll inv_count = 0;

    i = left;
    j = mid;
    k = left;
    while ((i <= mid - 1) && (j <= right)) {
        if (arr[i] <= arr[j])
            temp[k++] = arr[i++];
        else {
            temp[k++] = arr[j++];
            inv_count = inv_count + (mid - i);
        }
    }
    while (i <= mid - 1)
        temp[k++] = arr[i++];
    while (j <= right)
        temp[k++] = arr[j++];
    for (i = left; i <= right; i++)
        arr[i] = temp[i];
```

```
    return inv_count;
}
```

2 Graphs

2.1 Depth First Traversal (base)

```
ll dfs(int node, vector<int> adjacency[], bool visited[]) {
    visited[node] = true;
    for(auto i : adjacency[node])
        if(!visited[i]) dfs(i, adjacency, visited);
}
```

2.2 Breadth First Traversal (base)

```
queue<int> tovisit;
ll bfs(bool visited[]) {
    while(!tovisit.empty()) {
        visited[tovisit.front()] = true;
        for(int i : adjacency[tovisit.front()])
            if(!visited[i]) tovisit.push(i);
        tovisit.pop();
    }
}
```

3 Flows

Idk any flow algorithms/ideas yet, so this section is empty.