Homework 2 Solutions

1. Applying the master method, we have

   a. $4 > 4^0$, so $C(n) \in \Theta(n^{\log_4 4}) = \Theta(n)$;

   b. $4 = 4^1$ (since $\lim_{n\to\infty} \frac{n - \lg n}{n} = 1$), so $C(n) = \Theta(n^1 \lg n) = \Theta(n \lg n)$;

   c. $4 < 4^2$, so $C(n) = \Theta(n^2)$.

2.

```
// compute frequency for each element and return the one with the highest frequency
template <class Item>
Item brute-mode(vector<Item> v, int lo, int hi)
{
     vector<int> freq;
     for (int i = lo; i <= hi; ++i)
     {
         int count(0);
         for (int j = lo; j <= hi; ++j)
             if (v[j] == v[i])
                 ++count;
         freq.push_back(count);
     }

     int indexMax(0);
     for (int i = 1; i < freq.size(); ++i)
         if (freq[i] > freq[indexMax])
              indexMax = i;
     return v[lo + indexMax];
}
```

   The number of array element comparisons is $\sum_{i=\text{lo}}^{\text{hi}} \sum_{j=\text{lo}}^{\text{hi}} (1) = (\text{hi} - \text{lo} + 1)^2 = n^2$.

3.

```
// find the mode of m = v[lo..hi-1].  The answer is either m or v[hi]
template <class Item>
Item decrease-mode(vector<Item> v, int lo, int hi)
{
    if (lo == hi)
        return v[lo];
    Item m = decrease-mode(v, lo, hi-1);
    int f_m(0), f_hi(0);
    for (int i = lo; i <= hi; ++i)
    {
        if (v[i] == m)
            ++f_m;
        if (v[i] == v[hi])
            ++f_hi;
    }
    if (f_m > f_hi)
        return m;
```

```
    else
        return v[hi];
}
```

Again, we count the number of array element comparisons:

$$\begin{aligned} C(1) &= 0 \\ C(n) &= C(n-1) + \sum_{i=\mathrm{lo}}^{\mathrm{hi}} (2) \\ &= C(n-1) + 2(\mathrm{hi} - \mathrm{lo} + 1) \\ &= C(n-1) + 2n,\ n > 1. \end{aligned}$$

Hence

$$\begin{aligned} C(n) &= \sum_{i=2}^{n} 2(i) \\ &= 2\left(\frac{n(n+1)}{2} - 1\right) \\ &= n(n+1) - 2 \end{aligned}$$

4.

```
// sort and then find the length of the largest block of some value
template <class Item>
Item transform_mode(vector<Item> v, int lo, int hi)
{
    merge_sort(v, lo, hi);

    int mode_count(1), current_count(1);
    Item mode_value(v[lo]), current_value(v[lo]);

    for (int j = lo+1; j <= hi; ++j)
    {
        if (v[j] == current_value)
            ++current_count;
        else
        {
             if (mode_count < current_count)
             {
                   mode_count = current_count;
                   mode_value = current_value;
             }
             current_count = 1;
             current_value = v[j];
        }
    }
    if (mode_count < current_count)
        mode_value = current_value;
    return mode_value;
}
```

The number of array element comparisons is $\Theta(n \lg n) + \Theta(n) = \Theta(\mathrm{nlg}\, n)$.