School of Computing and Information Systems

COMP10002 Foundations of Algorithms

Sample Solution to Sample Mid-Semester Test #3

Question 1 (7 marks).

Yes, this is a pretty tough question to ask of twenty-odd minutes in a test. Its definitely harder than the one I had in the first sample test and hence was marked more leniently. Cross your fingers that I don't ask for something this complex in this year's test.

```
/* function to compute the mode of an integer array */
int
compute_mode(int A[], int n, int *mode) {
    int i, j, mloc=-1, mfrq=0, frq, dupes=0;
    if (n==0) {
        return MODE_UNDEFINED;
    }
    for (i=0; i<n; i++) {
        /* see if A[i] is a new item that can be the mode */
        frq = 0;
        for (j=i; j<n; j++) {
            /* by counting equal values to the right */
            frq += (A[i]==A[j]);
        }
        if (frq>mfrq) {
            /* yes, a new mode is found, so track it */
            mloc = i;
            mfrq = frq;
            dupes = 0;
        } else if (frq==mfrq) {
            /* a tie with a previous mode, so neither useful */
            dupes = 1;
        }
    }
    if (dupes) {
        /* no clear winner found */
        return MODE_UNDEFINED;
    } else {
        /* have a clear winner, so store it using the pointer */
        *mode = A[mloc];
        return MODE_DEFINED;
    }
}
```

Question 2 (3 marks).

Looking back (the first question here was from the 2013 test, and this question was from the 2014 test), there were some hard ones in here too! Must have been a tough year.

(a) $f_1(n) + f_2(n) = 3n^2 + 10n\sqrt{n} + 2n^2 \log\log n \in O(n^2 \log\log n)$.

(b) $f_1(n) \times \sqrt{n} + f_2(n) \times \log n = 3n^{2.5} + 10n^2 + 2n^2 \log n \log\log n \in O(n^{2.5})$ and is the fastest growing function.

(c) $f_1(n) \times \log n - 1.5 f_3(n) = 3n^2 \log n + 10n^{1.5} \log n - 3n^2 \log n - 1.5(\log n)^2 \in O(n^{1.5} \log n)$

(d) $f_3(n)/f_2(n) = \frac{2n^2 \log n + (\log n)^2}{2n^2 \log\log n} \in O(\log n/ \log\log n)$, and is the slowest growing function.