

## Lab 2 Complexity

1. What is the best and worst performance for the following code in big O notation? Can you modify the code and let the best performance to be  $O(1)$ .

```
boolean bSearch(int f[], int n, int x){
    int j = 0;
    int k = n;
    while (j + 1 != k){
        int i = (j + k)/2;
        if (x >= f[i]){
            j = i;
        } else {
            k = i;
        }
    }
    return (f[j]==x);
}
```

2. SelectSort: What is the best, worst and average performance for the following code in big O notation?

```
void selectionSort(int f[], int n) {
    int i, j;
    for (i = 0; i < n; i = i + 1) {
        int k = i;
        for (j = i + 1; j < n; j = j + 1) {
            if (f[j] < f[k]) {
                k = j;
            }
        }
        int temp = f[i];
        f[i] = f[k];
        f[k] = temp;
    }
}
```

3. Find out the big O notation for the following running time.

- $5n^4 + 3n^3 + 2n^2 + 4n + 1$
- $10n^3 + 4n + 120$
- $4\log(n) + n^2 + 7$
- $n \cdot \log(n) + 8$

4. Reduce the following big O notation

- $O(n^4) + O(3)$
- $O(n^2) + O(n^2) + O(n) + 9$
- $O(\log(n)) + O(n) + O(1)$

5. What is the worst performance for the following code in big O notation? How many times will be currentMax updated for a randomly ordered array?

```
double arrayMax(double[ ] data) {  
    int n = data.length;  
    double currentMax = data[0];  
    for (int j=1; j < n; j++)  
        if (data[j] > currentMax)  
            currentMax = data[j];  
    return currentMax;  
}
```