CS4335 Tutorial 8

Question 5a.

```
minDiff() {
        int min = positive infinity;
        for (int i = 1; i <= x.size; i++)
                if(abs(x[i]-1 < min)
                        min = x[i]-1;
        return min;
}
```

Question 5b.

```
minDiff (int left, int right) {
        if (right – left == 0 || right – left == 1) return min (x[left] – left, x[right] - right);
        int mid = floor ((left + right) / 2);
        if (x[mid] > mid) return minDiff (left, mid);
        else return minDiff (mid, right);
}
minDiff (1, x.size);
```

Question 5c.

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + 1 \\ &= T\left(\frac{n}{2^2}\right) + 1 + 1 \\ &= T\left(\frac{n}{2^k}\right) + k \end{aligned}$$

$$\begin{aligned} \frac{n}{2^k} &= 1 \\ k &= \log_2 n \end{aligned}$$

$$\begin{aligned} T(n) &= T(1) + \log_2 n \\ &= 1 + \log_2 n \\ T(n) &= O(logn) \end{aligned}$$

Question 6a.

```
maxNumberOfSquaredIntervals(int[] X) {
	for (int i = 1; i <= X.size; i++) {
		find the shortest squared interval Y end at i;
		if (Y.size == 0) return null;
	}
	find max number of compatible Y intervals n with interval scheduling algorithm;
	return n;
}
```

Question 6b.

Running time at worst case:

$$O(n^2) + O(n)$$
$$= O(n^2)$$

Question 6c.

Let A = maximal set of non-overlapped squared interval.

If A has interval X starts at time i.

Replace X with another interval Y starts at time i.

A’s optimality is unchanged.

For each I, A can contain at most one interval starting at i.

Thus, the problem is equivalent to the interval scheduling problem and the algorithm is correct.