Prof. Emdad Khan

September 2019
Lab#1

Group 1

Group members:
Asad Ali Kanwal
Aser Ahmad Ibrahim Ahmad
Jean Wilbert Folsy
Zayed Hassan

Math Problem 1

To know if a function $f(x)$ is increasing/eventually decreasing

- Get its derivative, $\acute{f}(x)$.
- Get the zero of the derivative x_0, $where\ f'(x) = 0$.
- Test $f'(x)$ before and after x_0.

1) $f(x) = -x^2$

   $f'(x) = -2x = 0 \rightarrow x_0 = 0 \rightarrow (1)$

   Test $f'(x)$ before and after x_0:

   $let\ x = 1 \rightarrow f'(x) = -2(1) = -2 \rightarrow negative\ (2)$

   $let\ x = -1 \rightarrow f'(x) = -2(-1) = 2 \rightarrow positive\ (3)$

   From (1), (2), and (3):
   $f'(x)\ is\ increasing\ for\ the\ range\ [-\infty, 0], and\ decreasing\ for\ the\ range\ [0, \infty]$.

2) $f(x) = x^2 + 2x + 1$

   $f'(x) = 2x + 2 = 0 \rightarrow x_0 = -1 \rightarrow (1)$

   Test $f'(x)$ before and after x_0:
   $let\ x = -2 \rightarrow f'(x) = -2 \rightarrow negative\ (2)$

   $let\ x = 0 \rightarrow f'(x) = 2 \rightarrow positive(3)$

   From (1), (2), and (3):
   $f'(x)\ is\ decreasing\ for\ the\ range\ [-\infty, -1], and\ increasing\ for\ the\ range\ [-1, \infty]$.

3) $f(x) = x^3 + x = x(x^2 + 1)$

   $f'(x) = 3x^2 + 1$

   We note that for any value of x, $f'(x)$ will be positive. This means that
   $f(x)\ is\ increasing\ for\ [-\infty, \infty]$.

Math Problem 2

To know how a function $f(x)$ grows with respect to another function $g(x)$, we divide them:

- If $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 0$, then $f(x)$ asymptotically grows no faster than $g(x)$.
- If $\lim_{x\to\infty} \frac{f(x)}{g(x)} = \infty$, then $g(x)$ asymptotically grows no faster than $f(x)$.
- If $\lim_{x\to\infty} \frac{f(x)}{g(x)} = c\ , where\ c\ is\ a\ constant$, then $f(x)$ and $g(x)$ asymptotically grow with the same rate.

1) $f(x) = 2x^2, g(x) = x^2 + 1$

$$\lim_{x\to\infty}\frac{f(x)}{g(x)} = \lim_{x\to\infty}\frac{2x^2}{x^2+1} = \lim_{x\to\infty}\frac{2}{1+\frac{1}{x^2}} = \frac{2}{1+0} = 2$$

$\therefore f(x)\ and\ g(x)\ asymptotically\ grow\ with\ the\ same\ rate.$

2) $f(x) = x^2, g(x) = x^3$

$$\lim_{x\to\infty}\frac{f(x)}{g(x)} = \lim_{x\to\infty}\frac{x^2}{x^3} = \lim_{x\to\infty}\frac{1}{x} = \frac{1}{\infty} = 0$$

$\therefore f(x)\ asymptotically\ grows\ no\ faster\ than\ g(x).$

3) $f(x) = 4x + 1, g(x) = x^2 - 1$

$$\lim_{x\to\infty}\frac{f(x)}{g(x)} = \lim_{x\to\infty}\frac{4x+1}{x^2+1} \to Apply\ L'Hopital's\ rule:$$

$$\lim_{x\to\infty}\frac{4}{2x} = \frac{4}{\infty} = 0$$

$\therefore f(x)\ asymptotically\ grows\ no\ faster\ than\ g(x).$

Problem 1: GCD Algorithm

```
private static int gcd(int m, int n) {
        int result = 1;
        for (int i = 1; i <= m && i <= n; i++) {
            if (m % i == 0 && n % i == 0)
                result = i;
        }
        return result;
    }
```

Problem 2: Subset sum problem – Brute Force Solution

```
private static int[] SubsetSum(int[] S, int k) {
        Arrays.sort(S);

        int[] T = new int[0]; //this array will hold the required subset
        int size = 0; //the final size of the required subset
        for (int i = S.length - 1; i >= 0; i--) { //scan the array from the end
            int temp = k;
            T = new int[S.length];
            int TSize = 0;
            for (int j = i; j >= 0; j--) {
                if (temp - S[j] >= 0) {
                    temp -= S[j];
                    T[TSize] = S[j];
                    TSize++;
                }
```

```
            }
            //if we found the sum. We get the found subset size
            if (temp == 0) {
                size = TSize;
                break;
            }
        }
        //copy the found subset to a new array with exactly the found subset size
        int[] result = new int[size];
        for(int i=0 ; i<size ; i++){
            result[i] = T[i];
        }
        return result;
    }
```

Problem 3: Greedy strategies

```
private static int[] GreedySubsetSum(int[] S, int k) {
        Arrays.sort(S);

        int[] T = new int[0]; //initialize the subset
        int size = 0; //initialize the subset size
        for (int i = 0; i < S.length; i++) {
            int temp = k;
            T = new int[S.length];
            int TSize = 0;
            for (int j = i; j < S.length; j++) {
                if (temp - S[j] >= 0) {
                    temp -= S[j];
                    T[TSize] = S[j];
                    TSize++;
                }
            }
            if (temp == 0) {
                size = TSize;
                break;
            }
        }
        //copy the found subset to a new array with the exact size
        int[] result = new int[size];
        System.arraycopy(T,0,result,0,size);
        return result;
    }
```

Problem 4

The solution will always be true because if we remove last element from *S* (which is also an element of *T*)

then it will be equal to the subsetSum of K', where K' is *K – last element.* Also, we remove the element from T.

For example, S = {4, 5, 10, 12};

K = 22;

T = {10, 12};

Now, S' = {4, 5, 10};

K' = K – 12 = 10

T' = T – {12} = {10}

Here K' is the sum of T' subset.