Algorithm: Lab 1

Houssam Eddine ATIF

N# 610165

Problem 1:

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

$\Rightarrow f'(x) = -2x$

$\Rightarrow f'(x) \geq 0$ in $]-\infty, 0]$ and $f'(x) \leq 0$ in $[0, +\infty[$

$\Rightarrow$ ~~x~~ $f(x)$ is increasing in $]-\infty, 0]$ and decreasing in $[0, +\infty[$

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

$\Rightarrow f'(x) = 2x + 2$

$\Rightarrow f'(x) \leq 0$ in $]-\infty, -1]$ and $f'(x) \geq 0$ in $[-1, +\infty[$

$\Rightarrow$ ~~x~~ $f(x)$ is decreasing in $]-\infty, -1]$ and increasing in $[-1, +\infty[$

c- $f(x) = x^3 + x$

$\Rightarrow f'(x) = 3x^2 \Rightarrow$ wherever x $f'(x) \geq 0$

$\Rightarrow f(x)$ is increasing.

Problem 2:

a/ $4x^3 + x$ is $\Theta(x^3)$

$$\Rightarrow \lim_{x \to +\infty} \frac{4x^3 + x}{x^3} = \lim_{x \to +\infty} \left(4 + \frac{1}{x^2}\right) = 4 \quad \text{is finite} \Rightarrow \text{True}$$

b/ $\log(n)$ is $O(n)$

$$\Rightarrow \lim_{n \to +\infty} \frac{\log n}{n} = \lim_{n \to +\infty} \frac{\frac{1}{n} \log e}{1} \quad (\text{l'hopitale Theory})$$

$$= \lim_{n \to +\infty} \frac{\log(e)}{n} = 0 \quad \Rightarrow \text{True}$$

c/ 2^n is $\omega(n^2)$

we need to know the output of $\lim_{x \to +\infty} \frac{x^2}{2^x}$?

lets make $f(x) = \frac{x^2}{2^x}$

there is a rule : if $0 < \lim_{x \to +\infty} \frac{f(x+1)}{f(x)} < 1 \Rightarrow \lim_{x \to +\infty} f(x) = 0$

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

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

$$= \lim_{x \to +\infty} \frac{x^2 + 2x + 1}{2x^2}$$

$$= \lim_{x \to +\infty} \frac{1 + \frac{2}{x} + \frac{1}{x^2}}{2} = \frac{1}{2} < 1$$

$\Rightarrow \boxed{\lim_{x \to +\infty} f(x) = 0}$ which means that $\boxed{2^n \text{ is } \omega(n^2)}$

d/ 2^n is $o(3^n)$

$$\lim_{x \to +\infty} \frac{2^n}{3^n} = \lim_{x \to +\infty} \left(\frac{2}{3}\right)^n \qquad \left(\frac{2}{3} < 1\right)$$

$$= 0$$

$\Rightarrow 2^n$ is $o(3^n)$ is True

Problem 3:

we need to prove that all $n>4$, $2^n < n!$: $\varphi(n)$

We will use totale induction methode

$\Rightarrow n=5 \Rightarrow 2^5 = 32$, $5! = 120$

$\Rightarrow 32 < 120$ it's true

$\Rightarrow$ Now under the Assumption that $n>5$ and that each of $\varphi(5), \varphi(5+1)$ ... $\varphi(n-1)$ are true, we will prove that $\varphi(n)$ is also true.

$\Rightarrow \varphi(n-1)$ is true $\Leftrightarrow 2^{n-1} < (n-1)!$

$\Leftrightarrow 2^{n-1} < (n-1)! < (n-1)!\frac{n}{2}$ $\quad (\frac{n}{2} > 1)$

$\Leftrightarrow 2^{n-1} < (n-1)!\frac{n}{2}$

$\Leftrightarrow \boxed{2^n < n!}$

$\Rightarrow$ So $2^n < n!$ for all $n>4$

Problem 4

```
public static int GCD (int a, int b) {
          int r;
          r = a % b;
          if (r == 0)
            return b;
          else
            return GCD (b, r);
        }
```

Problem 5

```
public static int secondSmallest(int[] arr) {

            int m1;
            int m2;

            if(arr==null || arr.length < 2) {
            throw new IllegalArgumentException("Input array too small");
            }

            if(arr[0]>arr[1]) {
                  m1=arr[1];
                  m2=arr[0];
            }else {

                  m2=arr[1];
                  m1=arr[0];
            }

            for(int i=2;i<arr.length;i++) {
                  if(m1>arr[i]) {
                        m2=m1;
                        m1=arr[i];
                  }
                  else if(m1<arr[i] && m2>arr[i]) m2=arr[i];

            }
            return m2;
      }
```

Problem 6

```java
public static int sum(List<Integer> b) {
			return b.stream().mapToInt(Integer::intValue).sum();
	}


	public static List<Integer> SubsetSum(int[] arr, int k) {

		int s;
		List<List> P=new ArrayList<>();
		List<Integer> S=new ArrayList<>();

		P.add(S);
		List<Integer> T=new ArrayList<>();
		if(k==0) return S;
		for(int i=0;i<arr.length;i++){

			s=P.size();
			for(int j=0;j<s;j++) {
				T=new ArrayList(P.get(j));
				T.add(arr[i]);

				if(sum(T)==k) {
					return T;
				}
				P.add(new ArrayList(T));
				T.clear();
			}

		}
		return null;
	}
```