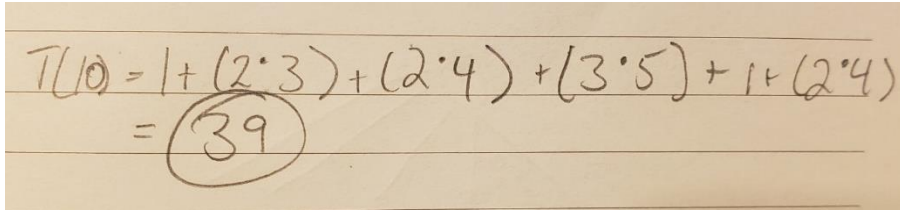


Homework 4 Question 3


$$T(10) = 1 + (2 \cdot 3) + (2 \cdot 4) + (3 \cdot 5) + 1 + (2 \cdot 4) \\ = 39$$

a)

b) We used an algorithm for computing  $T(10)$  that did not use any recursion but runs in  $O(n)$  time

\*Written in Python3

c)  $T = [1, 2, 3, 2, 4, 3, 5, 1, 4, 2]$

```
print(T)
```

```
n = len(T)
```

```
r = []
```

```
r.append(0)
```

```
for j in range(1,n):
```

```
    if j % 2 == 1:
```

```
        if j != n-1:
```

```
            val = 0
```

```
            if (T[j] == 1):
```

```
                if T[j-1] != 'X':
```

```
                    val = T[j-1] + 1 + r.pop()
```

```
            else:
```

```
                val = 1 + r.pop()
```

```
        elif(T[j-1] == 1):
```

```
            val = T[j-1] + (T[j] * T[j+1]) + r.pop()
```

```
            T[j+1] = 'X'
```

```
        elif (T[j-1] == 'X'):
```

```
            val = (T[j] * T[j+1]) + r.pop()
```

```
            T[j+1] = 'X'
```

```
        else:
```

```
            if (T[j-1] * T[j]) > (T[j] * T[j+1]):
```

```
                val = (T[j-1] * T[j]) + r.pop()
```

```
                T[j-1] = 'X'
```

```
            else:
```

```
                val = (T[j] * T[j+1]) + r.pop()
```

```
        T[j+1] = 'X'
    r.append(val)
else:
    val = 0
    if T[j-1] != 'X':
        val = max(T[j-1] * T[j], T[j-1] + T[j]) + r.pop()
    else:
        val = T[j] + r.pop()
    r.append(val)
elif j == n-1:
    val = T[j] + r.pop()
    r.append(val)
print(r[0])
```