# Homework 4

Robbie McKinstry, Jack McQuown, Cyrus Ramavarapu

9 September 2016

## Greedy Problems

### Problem 12:

### Problem 18:

## Dynamic Programming

### Problem 1:

#### A:

#### B:

To show that only $\mathrm{O}\big(n^2\big)$ operations are needed if every duplicate $T(i)$ is calculated only once, begin by expanding the sum in the recurrence.

$$T(n) = \sum_{i=1}^{n-1} T(i)T(i-1)$$

$$T(n) = T(1)T(0)+T(2)T(1)+T(3)T(2)\cdots+T(n-2)T(n-3)+T(n-1)T(n-2)$$

Since every $T(i)$ will only be calculated once, following sequence can be observed by counting the number of operations needed to determine each $T(i)$.

| T(i) | T(2) | T(3) | T(4) | T(5) | T(6) |
|---|---|---|---|---|---|
| Ops | 1 | 3 | 5 | 7 | 9 |

It can be shown that the $T(i+1)$ element of the sum requires two additional operations to calculate: *a multiplication and an addition.* Hence, this sequence will continue. It can be proven inductively that a closed form expression for the sum of operations required is $n^2$. Therefore, in this case $\mathrm{O}\big(n^2\big)$ operations are required.

**C:**

A $\mathrm{O}\big(n\big)$ algorithm can be derived from the original recurrence relationship by first eliminating the summation by calculating $T(n+1)$ in the following manner.

$$T(n+1) = \sum_{i=1}^{n} T(i)T(i-1)$$

$$T(n) = \sum_{i=1}^{n-1} T(i)T(i-1)$$

$$T(n+1) - T(n) = \sum_{i=1}^{n} T(i)T(i-1) - \sum_{i=1}^{n-1} T(i)T(i-1)$$

$T(n+1)$ and $T(n)$ overlap for all values $i : 1 \leq i \leq n-1$, therefore subtracting the two sums leaves only the the final in the sum for $T(n+1)$.

$$T(n+1) - T(n) = T(n)T(n-1)$$

The values for n can be shifted by setting $n = m - 1$.

$$T(m) - T(m-1) = T(m-1)T(m-2)$$

However, the label $m$ is without meaning, so label $m = n$.

$$T(n) - T(n-1) = T(n-1)T(n-2)$$

Equivalently,

$$T(n) = T(n-1)[1 + T(n-2)]$$

This expression is easily expressed as a single $\mathrm{O}\big(n\big)$ loop.

$Array: \; T$
$T[0] = 2$
$T[1] = 2$
**for** $i \leftarrow 2$ *to* $n$ **do**
$\quad | \quad T[i] = T[i-1] * (1 + T[i-2])$
**end**
$Output: \; T[n]$