Jake Traut
103134737

CSCI 3104 Homework 7

1. Yuckdonald's problem: dynamic programming

Start by defining valid locations, $location(m_i, m_j) = \begin{cases} 0 \text{ if } m_i - m_j < k \\ 1 \text{ if } m_i - m_j \geq k \end{cases}$
checking if location m_i is in valid distance k from m_j, the previous location
Then can define the maximum profit at location i as

$$P[i] = \max \begin{cases} \max_{j<i} \{P[j] + location(m_i, m_j) \cdot p_i\} \\ p_i \end{cases}$$

Thus the max expected profit at location i takes into account the maximum profits of previous locations j. Algorithm goes as follows:

```
max-profit(n, P)  //n locations, P[1,...,n] with profits at location i∈[1,n]
  profit[] // initialize array for max expected profit, will be output
  for i=1 to n:
     profit[i]=0
  for i=2 to n:
     for j=1 to i-1:
        prof = profit[j] + location(m_i, m_j)·P[i]
        if prof > profit[i]:
           profit[i] = prof
     if profit[i] < P[i]:
        profit[i] = P[i]
```

The double for loops in the algorithm gives a running time of $O(n^2)$

2. Palindromic, sequence $x[1..n]$

Consider the integers $1 \leq i < j \leq n$ for the subsequence $x[i..j]$ where $L(i,j)$ is the length of the longest palindrome.

Base case for $i = j$ set $L(i,j) = 1$ for a palindrome of one character and length 1.

If $i > j$ then $L(i,j) = 0$ for empty or invalid sequence

Else if $x[i] == x[j]$:

$$L(i,j) = \max\{L(i+1,j), L(i,j-1), L(i+1,j-1)+2\}$$

Else:

$$L(i,j) = \max\{L(i+1,j), L(i,j-1)\}$$

Honor Code Pledge: