Homework 1. Part 2
Due: Monday, January 15, 2018 before 8am EST.

Problem 1 [DPV] 6.2 – Hotel stops with minimum penalty.

(a) Define the entries of your table in words. E.g., $T(i)$ or $T(i,j)$ is ....

Let p(i) be the minimum penalty on $a_1, a_2, ..., a_i$ where a_i is included.

(b) State recurrence for entries of table in terms of smaller subproblems.

$p(i) = \min_j\{p(j) + (200 - (a_i - a_j))^2\}$ where $j < i$

(c) Write pseudocode for your algorithm to solve this problem.

Input : an array a of size N
Output: the minimum penalty by taking the optimal path of hotels

initialize array p of size N to 0;
for $i \in 1 \longrightarrow N$ **do**
  $min\_j = i - 1$;
  for $j \in 0 \longrightarrow i - 2$ **do**
    if $p[j] + (200 - (a[i] - a[j]))^2 < p[min\_j] + (200 - (a[i] - a[min\_j]))^2$ **then**
      $min\_j = j$
    end
  end
  $p[i] = p[min\_j] + (200 - (a[i] - a[min\_j]))^2$
end
return *last element in array p*

(d) Analyze the running time of your algorithm.

Since there are two for loops (one looping through N items and the other going back to find the minimum value), it will run in $O(n^2)$

Problem 2 [DPV] 6.3 – Yuckdonald's

(a) Define the entries of your table in words. E.g., $T(i)$ or $T(i, j)$ is ....

Let z(i) be the maximum profit in $\begin{cases} m_1, m_2, ..., m_i \\ p_1, p_2, ..., p_i \end{cases}$ where m_i and p_i are included.

(b) State recurrence for entries of table in terms of smaller subproblems.

$z(i) = p_i + \mathrm{max}_j\{z_j : (m_i - m_j) < k\}$

(c) Write pseudocode for your algorithm to solve this problem.

```
Input  : an array m and array p of size N and minimum distance k
Output: the maximum profit gathered from optimal store placement

initialize array z of size N to 0;
for i ∈ 1 ⟶ N do
    max_j = 0 ;
    for j ∈ 0 ⟶ i do
        if p[j] > p[min_j] then
            max_j = j
        end
    end
    if max_j != None and (m[i] − m[j]) >= k then
        z[i] = p[i] + p[max_j]
    else
        z[i] = p[i]
    end
end
max_profit = 0 for i ∈ 1 ⟶ N do
    if z[i] > z[max_profit] then
        max_profit = i
    end
end
return return z[max_profit]
```

(d) Analyze the running time of your algorithm.

Since there are two for loops (one looping through N items and the other going back to find the minimum value), it will run in $O(n^2)$

Problem 3 [DPV] 6.17 – Coin changing (unlimited supply of each denomination) Practice Only

(a) Define the entries of your table in words. E.g., $T(i)$ or $T(i, j)$ is ....

(b) State recurrence for entries of table in terms of smaller subproblems.

(c) Write pseudocode for your algorithm to solve this problem.

(d) Analyze the running time of your algorithm.

Problem 4 [DPV] 6.18 – Coin changing (use each denomination at most once) Practice Only

(a) Define the entries of your table in words. E.g., $T(i)$ or $T(i, j)$ is ....

(b) State recurrence for entries of table in terms of smaller subproblems.

(c) Write pseudocode for your algorithm to solve this problem.

(d) Analyze the running time of your algorithm.