1 Homework 10

Name: Kangdong Yuan

1.1 problem1

a).I did not work in a group.
b).I did not consult without anyone my group members
c).I did not consult any non-class materials.

1.2 problem2

Create an array B of length n and put 0 in each entry. For each element a $\in$ A , add 1 to B[a.deadline]. For each element in B, if B[a.deadline] $>$ a.deadline, return False (set is not independent). Otherwise, we continue the for loop. If there is no B[a.deadline] $>$ a.deadline in whole for loop return True (set is independent). The time complexity of this algorithm is $O(|A|)$.

1.3 problem3

Let OPT(j) be the maximum number of missiles that can be destroyed for the interval $[x_1, x_2, ...x_j]$. If the input ends at x_j , , so the choice is just when to last activate it before step j. Thus OPT(j) is the best of these choices over all i.

$OPT(j) = max_{o \leq i \leq j}\ [OPT(i) + min\{x_j, f(j-i)\}]$

set OPT(0) = 0
for j = 1 to n
$\quad OPT(j) = max_{o \leq i \leq j}\ [OPT(i) + min\{x_j, f(j-i)\}]$
endfor
return OPT(n)
The running time is O(n) per iteration, for a total of $O(n^2)$.

1.4 problem4

a).
This greedy approach will not be optimal, it will not give us best solution. For example sequence (2,20,2,2,1,1). If the first player use greedy approach

and takes the first card with a value of 2 then the second player can take the card witha value of 20 and win.
The better solution the first player is to take the last card with a value of 1. Then the second player will take either the 2 or the remaining 1 and the first player can take 20.

b).
we define that OPT(i,j) be the difference between, i is the largest total score first player can obtain, j is corresponding score of the second player, on the sequence interval s_i *to* s_j. And $V[i]$ is the value of each card, n is number of card in original sequence.

The Pre-computation code is
Pseudo-code
set the inital value
for i = 1...n:
    OPT[i, i] = v[i]

for j from i to n:
    for i from j to 1:
        OPT[i, j] = max (v[i] - OPT(i+1,j), v[j] - OPT(i,j-1))
return OPT array

The look up process:
choose s_i (first in squence), if $OPT[i, j] = v[i] - OPT(i + 1, j)$
choose s_j (last in squence), otherwise

The time complexity for Pre-computation is $O(n^2)$, the time complexity for each lookup is $O(1)$.