Q1

```
Max-Weight (T):
    if n = 1:
        return w
    else:
        sub = ⌊n/2⌋
        w1 = Max-Weight (T[1...sub])
        w2 = Max_Weight (T[sub... n])
        return max(w, w+w1, w+w2)
```

Procedure Max_Weight runs in time $\Theta(n)$. because worse case run time satisfy $T(n) = 2T(n/2) + \Theta(1)$

So. algorithm runs in $O(n)$ time.

Date . .

Let $G = (i_1, i_2, i_3 \ldots i_n)$

$G^* = (i_1^*, i_2^*, i_3^* \ldots i_n^*)$

where we assume G^* is the optimum solution.

Since G^* is optimal, it must contain at least many guards as G.

Lets assume the first j-1 elements are the same.

which means, $i_1 = i_1^*, i_2 = i_2^* \ldots i_j = i_j^*$.

but $i_j \neq i_j^*, i_{j+1} \neq i_{j+1}^* \ldots$

$G (\ldots\ldots i_j, i_{j+1} \ldots i_n)$

$G^* (\underbrace{\ldots\ldots}_{\text{same}} i_j^*, i_{j+1}^* \ldots i_n^*)$

since at moment of i_{j-1}, $t = f_{i_{j-1}^*}$, so according to algorithm, i_j is the guard with the greatest ending point. so we can replace i_j^* with i_j without messing around G^*. since $i_{j+1} - i_j > i_{j+1}^* - i_j$

$G' (\ldots\ldots i_j, i_{j+1}^* \ldots i_n^*)$

Inductive Step: Let $m \in \mathbb{N}$ and assume $P(m)$ holds.

$P(m)$: we can replace m leftmost guardians in G and get G' and G' is still the optimal solution.

WTP: $P(m+1)$ holds.

By IH, we suppose G' replaced m guardians. thus there's no solution to find less replaced gardian less than $m+1$. Hence $m+1$ guardians has been matched. $P(m) \Rightarrow P(m+1)$.

Therefore, we will eventually convert G into G^* without increasing number of guardians.

b). Sort the guards in ascending order of f_i.
and then loop through each ~~ga~~ guardian to find which one should be added into G. The sorting algorithm take $O(n\log n)$.

Q3:

(a). ~~Let array B be an array such that $0 \le i \le n$ where $B[i] = [A[i], A[i+1], A[i-1]]$~~
Let array A be an array such that $0 \le i \le n$ where $A[i]$ = True if
$|A[i+1] - A[i]| \le 1$, $|A[i-1] - A[i]| \le 1$

b).