1)

0, 1, 2, 3

a) $[s, f] = [0, 3]$

$S = \{[0,2], [2,3], [1,2], [0,1]\}$

$US = [0,3]$ ✓

Por la estrategia codiciosa presentada, se utilizarian los siguientes intervalos $[0,2], [0,1], [1,2], [2,3] \rightarrow N=4$

Pero realmente el optimo (menor numero de intervalos) seria tomar $[0,2][2,3]$. $\rightarrow N=2$

Por lo que se ve claramente que la estrategia codiciosa no es optima.

PASO A PASO

COND: minimo "si" posible

INT: $[0,2]$

COND: solapado y minimo "si" posible

INT: $[0,1]$

COND: solapado y minimo "si" posible

INT: $[1,2]$

COND: solapado y minimo "si" posible

INT: $[2,3]$

b) CubrimientoOptimo (s, f, S, L):

```
    max = in S
    for interval in S:                →   O(|S|)
        if interval[0] ≤ s:
            if interval[1] > max:
                max = interval[1]
                max_interval = interval

    L.append(max_interval)
    if max_interval[1] = f:
        return L
    else:
        CubrimientoOptimo(max_interval[1], f, S, L)
```

c) La complejidad base del algoritmo (sin contar el llamado recursivo) es de $O(|S|)$.

El peor caso es tener para $[0, n]$, n intervalos de $\Delta 1$, con lo que quedaría $O(n \cdot |S|)$

El mejor caso es tener para $[0,n]$, el intervalo $[0,n]$, donde quedaría una complejidad de $O(|S|)$.