Problema01:

Como todos valen lo mismo, se organizan de menor peso a mayor, para escoger luego según en el orden de menor peso hasta que se llene la capacidad. Asi se garantiza que siempre se obtenga la mayor cantidad de dinero posible.

Problema02:

El siguiente ejemplo es en base a hora:

0-2 (weitgh 25), 2-5(weitgh 25),5-7 (weitgh 25), 0-7 (weitgh 76),

cogería las 3 primeras actividades por endtime y dejaría la última porque chocaría con las otras y como es una no la tomara en cuenta antes de las 3 pero esta tiene más weitgh que las otras por eso fallarían el criterio greedy.

Problema 03:

Evaluamos la N gasolinera y obtenemos la mayor distancia posible de la gasolinera más lejana (guardamos el id de la posición y la distancia y además obtenemos tambien la posición de la última gasolinera para evaluarlo al final);

Tomamos en cuenta entre una y otra que la rodean puede ser a izquierda o derecha se toman en cuenta ambas, ya luego de recorrer ese for en tiempo N,

Luego con esa distancia si existe una gasolinera que tenga una distancia intermedia (mayor que el doble de la distancia más larga , pero que sus cercanas de ellas están relativamente cerca, se toma en cuenta esa posición media y se ubica a ahí la gasolinera del input)

Si no existen distancias intermedias el doble de la más larga se procede a ubicar en la ultima gasolinera a una distancia mayor que la mayor distancia obtenida puede ser hasta con más 1 (el caso es que sea un chin mayor y se ubica en esa posición) si la última posición está cerca de M y la distancia de la mayor evaluada en el for la sobre pasa (o excede) entonces no se puede poner la gasolinera a una distancia mayor por lo tanto se termina la ejecución). La complejidad de este algoritmo es O(N).

Problema 04:

Para este problema no existe una solución greedy ya que hay que iterarlo y tomar en cuenta cada sub String y ya esto sería hacer una especie de bruteforce.