

Algoritmos y Estructuras de Datos II - 1º cuatrimestre 2019
Práctico 3 - Parte 1: Algoritmos voraces

1. Demostrar que el algoritmo voraz para el problema de la mochila *sin fragmentación* no siempre obtiene la solución óptima. Para ello puede modificar el algoritmo visto en clase de manera de que no permita fragmentación y encontrar un ejemplo para el cual no halla la solución óptima.
2. Considere el problema de dar cambio. Pruebe o dé un contraejemplo: si el valor de cada moneda es al menos el doble de la anterior, y la moneda de menor valor es 1, entonces el algoritmo voraz arroja siempre una solución óptima.

Para cada uno de los siguientes ejercicios:

- Describa cuál es el criterio de selección.
- ¿En qué estructuras de datos representará la información del problema?
- Explique el algoritmo, es decir, los pasos a seguir para obtener el resultado. No se pide que "lea" el algoritmo ("se define una variable x", "se declara un for"), si no que lo explique ("se recorre la lista/el arreglo/" o "se elije de tal conjunto el que satisface...").
- Escriba el algoritmo en el lenguaje de programación de la materia.

-
3. Se desea realizar un viaje en un automóvil con autonomía A (en kilómetros), desde la localidad l_0 hasta la localidad l_n pasando por las localidades l_1, \dots, l_{n-1} en ese orden. Se conoce cada distancia $d_i \leq A$ entre la localidad l_{i-1} y la localidad l_i (para $1 \leq i \leq n$), y se sabe que existe una estación de combustible en cada una de las localidades.

Escribir un algoritmo que compute el menor número de veces que es necesario cargar combustible para realizar el viaje, y las localidades donde se realizaría la carga.

Suponer que inicialmente el tanque de combustible se encuentra vacío y que todas las estaciones de servicio cuentan con suficiente combustible.

4. En numerosas oportunidades se ha observado que cientos de ballenas nadan juntas hacia la costa y quedan varadas en la playa sin poder moverse. Algunos sostienen que se debe a una pérdida de orientación posiblemente causada por la contaminación sonora de los océanos que interferiría con su capacidad de inter-comunicación. En estos casos los equipos de rescate realizan enormes esfuerzos para regresarlas al interior del mar y salvar sus vidas.

Se encuentran n ballenas varadas en una playa y se conocen los tiempos s_1, s_2, \dots, s_n que cada ballena es capaz de sobrevivir hasta que la asista un equipo de rescate. Dar un algoritmo voraz que determine el orden en que deben ser rescatadas para salvar el mayor número posible de ellas, asumiendo que llevar una ballena mar adentro toma tiempo constante t , que hay un único equipo de rescate y que una ballena no muere mientras está siendo rescatada mar adentro.

5. Sos el flamante dueño de un teléfono satelital, y se lo ofrecés a tus n amigos para que lo lleven con ellos cuando salgan de vacaciones el próximo verano. Lamentablemente cada uno de ellos irá a un lugar diferente y en algunos casos, los períodos de viaje se superponen. Por lo tanto es imposible prestarle el teléfono a todos, pero quisieras prestárselo al mayor número de amigos posible.

Suponiendo que conoces los días de partida y regreso (p_i y r_i respectivamente) de cada uno de tus amigos, ¿cuál es el criterio para determinar, en un momento dado, a quien conviene prestarle el equipo?

Tener en cuenta que cuando alguien lo devuelve, recién a partir del día siguiente puede usarlo otro. Escribir un algoritmo voraz que solucione el problema.

6. Para obtener las mejores facturas y medialunas, es fundamental abrir el horno el menor número de veces posible. Por supuesto que no siempre es fácil ya que no hay que sacar nada del horno demasiado temprano, porque queda cruda la masa, ni demasiado tarde, porque se quema.
- En el horno se encuentran n piezas de panadería (facturas, medialunas, etc). Cada pieza i que se encuentra en el horno tiene un tiempo mínimo necesario de cocción t_i y un tiempo máximo admisible de cocción T_i . Si se la extrae del horno antes de t_i quedará cruda y si se la extrae después de T_i se quemará.
- Asumiendo que abrir el horno y extraer piezas de él no insume tiempo, y que $t_i \leq T_i$ para todo $i \in \{1, \dots, n\}$, ¿qué criterio utilizaría un algoritmo voraz para extraer todas las piezas del horno en perfecto estado (ni crudas ni quemadas), abriendo el horno el menor número de veces posible? Implementarlo.
7. Un submarino averiado descansa en el fondo del océano con n sobrevivientes en su interior. Se conocen las cantidades c_1, \dots, c_n de oxígeno que cada uno de ellos consume por minuto. El rescate de sobrevivientes se puede realizar de a uno por vez, y cada operación de rescate lleva t minutos.
- Escribir un algoritmo que determine el orden en que deben rescatarse los sobrevivientes para salvar al mayor número posible de ellos antes de que se agote el total C de oxígeno.
 - Modificar la solución anterior suponiendo que por cada operación de rescate se puede llevar a la superficie a m sobrevivientes (con $m \leq n$).
8. Vivís en la montaña, es invierno, y hace mucho frío. Son las 10 de la noche. Tenés una voraz estufa a leña y n troncos de distintas clases de madera. Todos los troncos son del mismo tamaño y en la estufa entra solo uno por vez. Cada tronco i es capaz de irradiar una temperatura k_i mientras se quema, y dura una cantidad t_i de minutos encendido dentro de la estufa. Se requiere encontrar el orden en que se utilizarán la menor cantidad posible de troncos a quemar entre las 22 y las 12 hs del día siguiente, asegurando que entre las 22 y las 6 la estufa irradie constantemente una temperatura no menor a $K1$; y entre las 6 y las 12 am, una temperatura no menor a $K2$.
9. (*sobredosis de limonada*) Es viernes a las 18 y usted tiene ganas de tomar limonada con sus amigos. Hay n bares cerca, donde cada bar i tiene un precio P_i de la pinta de limonada y un horario de happy hour H_i , medido en horas a partir de las 18 (por ejemplo, si el happy hour del bar i es hasta las 19, entonces $H_i = 1$), en el cual la pinta costará un 50% menos. Usted toma una cantidad fija de 2 pintas por hora y no se considera el tiempo de moverse de un bar a otro. Escribir un algoritmo que obtenga el menor dinero posible que usted puede gastar para tomar limonada desde las 18 hasta las 02 am (es decir que usted tomará 16 pintas) eligiendo en cada hora el bar que más le convenga.