1.– Aplique el algoritmo dado para el problema de la mochila en los siguientes casos:

a) M = 15, V = [10 5 15 7 6 18 3]

P = [2 3 5 7 1 4 1].

b) Seleccione un lote de 100 libros, lo más barato posible, de entre los siguientes lotes

Lote de 25	volúmenes	65.000 Ptas./Lote
Lote de 10	volúmenes	30.000 Ptas./Lote
Lote de 100	volúmenes	270.000 Ptas./Lote
Lote de 80	volúmenes	160.000 Ptas./Lote

2.– Construya procedimientos que resuelvan el problema de la **mochila**:

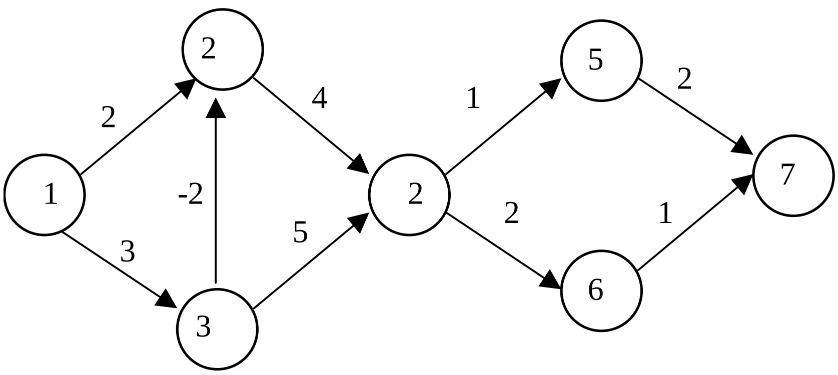
a) Ordenando primero los materiales (por precios).

b) Tomando los materiales que se supone se necesitarán. INDICACION: Una posible estimación de esta cantidad puede venir dada por

$$\frac{V}{\sum_{i=1}^{n} v_i}$$

3.– Plantee la función objetivo, para el problema de búsqueda de caminos óptimos con vértice común, como la suma de los arcos que forman los caminos de una solución factible. Diseñe un algoritmo devorador "inmediato" que obtenga una "buena" solución. Estime su calidad.

4.– Aplique el algoritmo de Dijkstra al siguiente grafo



para x_0 = 1. Busque otra solución mejor que la obtenida. ¿Por qué no se obtiene la solución óptima?

5.– Se quiere establecer una red de distribución de periódicos desde la ciudad A a otras 10 ciudades. Las distancias entre ellas en Kms. se da en el cuadro siguiente. Obtenga una red de distribución óptima para los casos en que

a) La distribución se hace directamente desde A a las demás ciudades, y

b) La distribución se va ramificando a través de las ciudades.

	A	B	C	D	E	F	G	H	I	J	K
B	40										
C	45										
D	50										
E	45										
F	30				40						
G		25				40					
H		60									
I			25	25				10			
J		50					20	10			
K	60			45	10						

6.– Implemente el **Algoritmo de Prim** para la obtención del árbol de expansión mínimo.

7.– Escriba un procedimiento Pascal que implemente el **Algoritmo de Huffman** para la obtención del **árbol extendido de mínimo peso**.

8.– **MEZCLA OPTIMA DE FICHEROS**.

Aplique el procedimiento anterior a la resolución del problema de la mezcla de N ficheros con tres canales: Tomamos dos ficheros X e Y, y devolvemos la mezcla de ambos en otro fichero Z. Cada fichero i tiene Q_i datos.

9.– Construya un procedimiento que decida si un grafo no dirigido dado es **conexo**.

10.– Un vértice p de un grafo dirigido G = (V,A) se dice que es un **pozo** o **sumidero** si hay arcos en el grafo que lo tienen como extremo y ninguno como origen. Diseñe un algoritmo que detecte la presencia de pozos en un grafo G.

11.– Un vértice p de un grafo dirigido G = (V,A) se dice que es un **manantial** o una **fuente** si hay arcos en el grafo que lo tienen como origen y ninguno como extremo. Diseñe un algoritmo que detecte la presencia de manantiales en un grafo G.

12.– **Planificación de Tareas con Caducidad**. Se tiene una serie de N trabajos cada uno de los cuales proporciona un beneficio b_i siempre y cuando se ejecute a lo más en un instante d_i, transcurrido el cual, su ejecución no proporcionaría ningún beneficio. En cada instante $t= 1,2,...,T$, se realiza un trabajo. Se trata de hallar los trabajos que hemos de realizar y la secuencia en que debemos hacerlos para obtener el máximo beneficio posible y calcular dicho beneficio. N y T son fijos.

13.– Los profesores de Metodología de la Programación no saben que hacer para tener una estimación de las notas de sus alumnos. En un momento de locura han creído descubrir un buen método para hacer esta estimación.

Puesto que por las prácticas realizadas en el laboratorio pueden tener una idea aproximada de los conocimientos de cada alumno, saben que **dado un problema *P*, si un alumno es capaz de resolverlo, emplea en su resolución un tiempo t_P**, no admitirán medias tintas, un problema estará o Bien o Mal resuelto según esté o no completamente resuelto. Se entiende asimismo que un alumno puede resolver un problema y otro no, aunque este último tenga en las prácticas una nota global superior.

El examen constará de M problemas cada uno de los cuales tendrá un peso de $1/M$ en la calificación global.

Se pide:

A) Si la dinámica del examen consiste en entregar un problema y proceder a su recogida transcurrido un tiempo, que dependerá del problema, y a continuación, proceder de la misma forma con el siguiente problema.

Dado un examen de M problemas calcular el número de aprobados esperado y los alumnos que superarían el examen junto con los ejercicios que resolverían.

B) Los alumnos han mostrado su disgusto por la poca libertad que les da el método de evaluación anterior, pidiendo que todos los problemas se les den al comenzar el examen y entregen su solución al final del examen. Si los profesores accediesen a su petición, calcular:

   i) el número máximo de aprobados,

   ii) el número mínimo de aprobados

En ambos casos debe también decirse cuáles son los alumnos que superarían el examen y justificar los algoritmos dados para i) y ii). Como en a) deben facilitarse los datos concernientes a cada alumno, esto es, cuál sería la secuencia de resolución de ejercicios propuestos en el examen.

C) Algunos alumnos han propuesto que se les dé el examen de antemano, a lo que los profesores, como era previsible, no han accedido. Pero han realizado una contraoferta, consistente en que no les darán los ejercicios, sino la estimación que han hecho sobre cada problema para cada alumno, esto es, el tiempo que necesita para resolverlo, y deben elegir entre N problemas los M que compondrán el examen. Remisos los alumnos a aceptar la contraoferta, los profesores han mejorado la misma añadiendo que en el examen se propondrán los N problemas entre los que se elegirán los ejercicios del examen, pero cada alumno habra de resolver sólo $M < N$. El examen tendrá una duración fijada por los profesores de antemano que también ha sido facilitada como dato a los alumnos. Calcular:

i) el número máximo de aprobados,

ii) el número mínimo de aprobados

Los datos que deben darse como solución a c) son los mismos que en b).

14.– Disponemos de un ordenador multitarea con N procesadores, cada uno de los cuales puede direccionar una cantidad de memoria de P_i Kbytes ($1 \leq i \leq N$). Queremos procesar N trabajos, cada uno de los cuales necesita m_i Kbytes de memoria. Evidentemente, para poder procesar un trabajo en un procesador, la cantidad de memoria que aquel necesita debe ser a lo sumo la que éste puede direccionar, además alguno de los trabajos es posible que no pueda ser procesado. Un procesador sólo puede realizar un trabajo del lote.

Diseñe un algoritmo voraz que asigne cada trabajo a un procesador, de forma tal que el número de trabajos que no pueden ser procesados sea mínimo. Pruébese que el algoritmo diseñado encuentra la solución óptima o encuéntrese un contraejemplo en caso contrario.

15.– **CUBRIMIENTO DE NODOS**.

Dado un grafo $G=(V,A)$ no dirigido, un subconjunto C de V es un "Cubrimiento de Nodos" si cada arista de A tiene al menos un extremo en C. Plantee el problema y diseñe un algoritmo devorador que obtenga un cubrimiento de un grafo dado, tratando de que tenga la menor cantidad posible de nodos. Estime la calidad de la solución.

16.– **Aplicaciones del cubrimiento de nodos**.

A) Una agencia de información tiene N sedes en varias ciudades del mundo. Cada sede puede comunicarse con otras (una o varias a la vez) de la misma agencia, para recibir o mandar información, a través de un terminal que en ella existe. Una red de información activa (RIA) es la que forman un número de terminales (nodos de la red) que en un determinado momento estén mandando o recibiendo información. Una arista de la red sería un flujo de información directa (FID) entre dos terminales. Cuando una RIA está constituida por los N terminales que existen puede producirse un colapso en la red, el cual solo se soluciona desconectando al menos uno de los dos extremos (terminales) de cada FID. El coste de este colapso lo mide el número de terminales que deben ser desconectadas. Se pide:

i) Plantee el problema y Diseñe un algoritmo devorador "rápido" (de complejidad no superior a cuadrática) que, dada una RIA (implementada de alguna forma que debe precisarse), determine qué terminales deben desconectarse para solucionar un colapso, intentando que el coste sea el menor posible. Supóngase que los terminales están numerados de 1 a N.

ii) Estime la calidad del algoritmo: Complejidad en tiempo y calidad de la solución (contraejemplo si no es óptima y demostración si lo es).

B) Para los juegos olímpicos son necesarios un total de *n* intérpretes (que deberán saber alguno de los *k* idiomas oficiales). De este grupo existirá un subgrupo que se encargará de recibir instrucciones generales, para transmitírselas a los demás intérpretes. Es deseable que este subgrupo sea lo más reducido posible. Se pide lo mismo que el caso A, para el problema de obtener un subgrupo de intérpretes, con las mismas restricciones de complejidad.

NOTA: Aunque si se resuelve el problema de obtener un subgrupo de intérpretes, lo más reducido posible, que sepa los k idiomas, el problema original queda resuelto, no plantearlo con este fin, ya que ofrece una solución demasiado trivial. Piénsese que si todos los intérpretes saben, por ejemplo, Inglés, bastaría con un sólo intérprete para formar el subgrupo deseado, sin que tenga que saber todos los idiomas.

17.– En la Alta Edad Media, y en una región dividida en *N* reinos, un intrigante Cardenal planea establecer una alianza internacional entre todos ellos basándose en una política matrimonial. Cada monarca tiene un único hijo y una o varias hijas, cada hijo tiene un determinado Peso. La planificación es la siguiente:

i) Se establece una alianza entre dos reinos si el hijo del rey de uno se casa con una hija del rey de otro.

ii) Si dos reinos están aliados, todos los aliados de uno también lo están con todos los aliados del otro.

iii) El peso de un matrimonio es la medida de su estabilidad.

iv) El Cardenal pretende unir todos los reinos consiguiendo la mayor estabilidad posible.

Diseñe un algoritmo devorador para resolver el problema del Cardenal. Calcule el orden de complejidad del mismo y probar si la solución obtenida es óptima.

18.– Se define el **diámetro** de un árbol binario como la longitud del camino más largo entre dos nodos cualesquiera del mismo. Desarrolle una estrategia voraz que calcule el diámetro de un árbol con una complejidad lineal con el número de nodos del árbol binario.

19.– Una sala ha de estar constantemente iluminada por un mínimo de *M* bombillas. Para ello, disponemos de un conjunto finito de *N* bombillas $\{b_i\}$ que una vez encendidas no pueden apagarse y que se gastan al cabo de t_i días. Se trata de maximizar el número de días que podemos tener la sala iluminada.

Resuelva el problema usando una estrategia voraz eficiente, justificándola. Calcule la complejidad del algoritmo desarrollado. Razone si la estrategia es óptima, y en caso contrario, exponga un contraejemplo.

20.– Partiendo de una sucesión *S* de *N* números enteros y un conjunto A de *N* números, reordene los elementos de A en una sucesión *S'* de manera que si se comparan los pares S_i y S_i', para todo $i \in \{1,...,N\}$, se verifica:

$$S_i \leq S_i' \text{ para un mayor número de veces posibles.}$$

Desarrolle una estrategia voraz para resolver el problema y prográmela.

NOTA: Sin pérdida de generalidad se puede suponer A ordenado en un vector. Por ejemplo:

A	0	1	1	2	3	8	10	11	12	13
S	1	3	7	10	9	2	4	17	14	10
S'	1	3	8	10	11	2	12	0	1	13

21.– **Cubrimiento de Conjuntos**. Sea $S = \{S_i\}_{i\in I}$, una colección finita de conjuntos finitos. Obténgase un recubrimiento mínimo de la misma.

Un **recubrimiento** de S es subcolección de la misma $S' = \{S'_j\}_{j\in J}$, (i.e. $\forall j\in J\ \exists\ i\in I,$, $S'_j = S_i$), tal que todo conjunto de S esté contenido en la unión de los conjuntos de S'.

Un **recubrimiento** S' de S se dice **mínimo** si el cardinal de S' es mínimo entre todos los recubrimientos de S, es decir, el número de conjuntos de la subcolección S' es mínimo.

Por ejemplo: Sea S = {{1,3,4}, {2,4,6}, {2,4}, {1,3}, {5,7}, {4,6}}. Recubrimientos del mismo son:

S' = {{1,3,4}, {2,4}, {5,7}, {4,6}},

S'' = {{1,3}, {2,4,6}, {5,7}}

S''' = {{1,3}, {2,4}, {5,7}, {4,6}}

Diseñe una **heurística voraz** para resolver el problema. Exponga un contraejemplo en el caso de que el algoritmo voraz no devuelva la solución óptima.

22.– En un campo tenemos un número determinado de flores, cada una con una determinada cantidad de polen. Tenemos un zángano que guiado por la ley del mínimo esfuerzo, sólo vuela a las flores adyacentes. Supuesto que debe saciar su apetito con una cantidad de polen *P*, visitando el menor número de flores, diseñe una **heurística voraz** que le ayude a conseguir su objetivo. Supóngase que parte de una flor dada.

¿Y si el zángano puede elegir la primera flor? Si quitamos la restricción de la elección a las flores adyacentes, ¿cuál sería la solución?

Calcule la complejidad del algoritmo desarrollado en cada uno de los casos.

23.– Tenemos dos bolsas capaces de soportar hasta un peso máximo *Pmax*, y un conjunto de *N* objetos de cada uno de los cuales conocemos su peso p_i. Se trata de decidir qué objetos se ponen en cada bolsa para que el peso en ambas esté lo más equilibrado posible, sin sobrepasar el peso máximo *Pmax*, sabiendo que el peso de los objetos de cada bolsa debe ser al menos *Pmin* (*Pmin* ≤ *Pmax*). Resuelva el problema mediante una **heurística voraz**, y demuestre la optimalidad de la estrategia o exponga un contraejemplo. Calcule asimismo la complejidad del algoritmo desarrollado.

24.– Un camionero que está en Babia quiere regresar a su casa, que está por los cerros de Úbeda, siguiendo una ruta dada y llevando un camión que le permite, con el depósito lleno, recorrer k kilómetros sin repostar. El camionero dispone de un mapa de carreteras que le indica las distancias entre las gasolineras que hay en su ruta. Como va con prisa, el camionero desea pararse a repostar el menor número de veces posible.

Diseñe e implemente un **algoritmo voraz** que nos diga en qué gasolineras tiene que parar. Demuestre que el algoritmo encuentra siempre la solución optima y calcule la complejidad de dicho algoritmo.

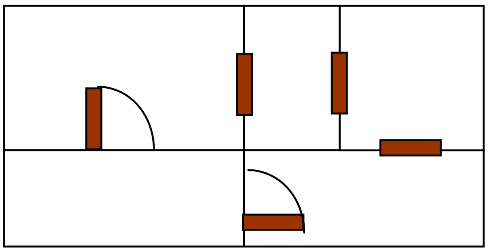
25.– Se considera una secuencia de palabras $p_1, p_2, ..., p_n$, de longitudes $l_1, l_2, ..., l_n$. Se desea agruparlas en líneas de longitud L. Las palabras están separadas por espacios cuya amplitud ideal (en milímetros) es b, pero los espacios pueden reducirse o ampliarse si es necesario (aunque sin solapamiento entre palabras), de tal forma que una línea $p_i, p_{i+1}, ..., p_j$ tenga exactamente longitud L. Sin embargo, existe una penalización por reducción o ampliación del número total de espacios que aparecen o desaparecen. El coste de fijar una línea $p_i, p_{i+1}, ..., p_j$ es $(j - i)|b' - b|$, siendo b' el ancho real de los espacios, es decir, $(L - l_i - l_{i+1} - ... - l_j)/(j - i)$. No obstante, si $j = n$ (la última palabra) el coste será cero a menos que $b' < b$ (ya que no es necesario ampliar la última línea).

Diseñe e implemente un **algoritmo voraz** que resuelva el problema. Demuestre la optimalidad de solución proporcionada o en caso contrario, exponga un contraejemplo.

26.– Para *Forrest Gump* la vida era como una caja de bombones. Cada caja tenía bombones de varias clases. *Forrest* frecuentaba una tienda que tenía muchas cajas distintas, y *Forrest* siempre deseó tener bombones de todas las clases, pero su paga no le alcanzaba. Por ello, tenía que comprar **el menor número de cajas** pues todas las cajas tenían el mismo precio. Diseñe una **heurística voraz** que le diga a *Forrest* qué cajas debe comprar y cuánto le costará comprarlas.

27.– Los *Reyes Magos* tienen sus encargos. Necesitan repartir un conjunto de N regalos voluminosos y de considerable peso. Para ello, pueden utilizar hasta M camellos, cada uno de los cuales puede soportar un peso dado (no es el mismo para todos los camellos). Los *Magos* deben de repartir la mayor cantidad posible de regalos. Se trata de decir cuál es la distribución de los objetos en los camellos, supuesto que en cada camello va un sólo objeto. Diseñe una ***estrategia voraz*** que resuelva el problema y calcule su complejidad.

28.– El plano de una vivienda puede verse como un conjunto de habitaciones rectangulares, que comparten paredes unas con otras, y que tienen una o más puertas para pasar de una habitación a otra. Se trata de calcular la temperatura que habrá en cada habitación partiendo de una temperatura inicial y de una determinada situación de cada puerta (abierta/cerrada). Diseñe un algoritmo que calcule la temperatura final de cada habitación, sabiendo que la temperatura de aquellas habitaciones que compartan una puerta abierta debe calcularse mediante una sencilla media aritmética ponderada al volumen de la habitación. Nótese que el número de habitaciones conectadas puede ser de dos o más.



29.– El concurso "*Los Glotones*", el sueño de los golosos, consiste en que cada equipo se coma, en un tiempo dado, la mayor cantidad de tartas. Cada componente del equipo debe comerse una tarta y sólo una tarta, tarta que solo se contabiliza si se la come completamente.

Al mismo se han dirigido los GVI (Glotones Voraces de Informática, GGP o G2P en terminología anglosajona), que han formado un equipo, pues les gustan más los pasteles que las cucarachas, y se han propuesto ganar el concurso aplicando sus conocimientos informáticos.

Para ello, han hecho una serie de pruebas para determinar la cantidad de tarta que cada uno es capaz de ingerir en el tiempo dado por el concurso. También han llevado a un ojeador, que además es el capitán del equipo, capaz de hacer una estimación, al instante, del peso de cada tarta. El capitán del equipo debe asignar qué componente del equipo se comerá cada tarta (las tartas están numeradas).

Diseñe el algoritmo que utilizará el capitán para realizar la asignación de los componentes del equipo a las tartas, para maximizar el número de tartas "devoradas" por completo. Calcule la complejidad del algoritmo, que debe ser rápido, porque el tiempo empieza a correr en cuanto se enseñan las tartas.

30.– ¡**Nos invaden**! El enemigo, armado hasta los dientes con palos y piedras ha desembarcado en las costas de nuestro país invadiendo N ciudades. Los servicios de "inteligencia" están informados que en cada una de las ciudades invadidas se encuentran e_i efectivos enemigos. Para contraatacar, el grupo de intervención rápida de defensa dispone de N equipos listos para intervenir. Cada uno de ellos consta de d_j efectivos completamente equipados y entrenados. Para garantizar el éxito de la intervención en una ciudad es necesario que contemos al menos con tantos efectivos de defensa como el enemigo.

Diseñe un **algoritmo voraz** que indique qué grupo de intervención debe ir a cada ciudad de forma que se maximice el número de éxitos garantizados. Asimismo, calcule la complejidad del algoritmo desarrollado.

31.– Eran principios de septiembre, fechas en las que suelen ser los exámenes de recuperación para recuperar las calabazas obtenidas durante el curso, calabazas entre las que por desgracia a menudo se encuentra Ampliación de Programación. Juan Jesús Penso, que se había metido a hortelano durante la estación estival, tenía “sembrada” su huerta de *N* agujeros de conejos que habían hecho sus madrigueras en su huerto. Además, los pájaros no hacían más que comerse sus hortalizas.

Un día Juan Jesús Penso, ideó un plan para matar dos pájaros de un tiro, y evitar el daño de tanto “animal”. Decidió que sería buena idea poner espantapájaros por todo su huerto para evitar que tanto los conejillos como los pajarillos le fastidiaran la cosecha. Tenía *M* estacas, cada una con un diámetro d_i, que aprovechando los agujeros de los conejos, metería en los mismos.

Sabiendo que cada estaca tiene un diámetro d_i y que, a su vez cada agujero tiene un diámetro a_i. Diseñe un **Algoritmo Voraz** que ayude a Juan Jesús Penso a colocar las estacas en los agujeros que se puedan meter, de tal forma que se maximice el número de agujeros en los que hay una estaca.

32.– La Universidad tiene que planificar un evento cultural que consiste en *N* conferencias. Para cada conferencia se conoce la hora de comienzo y la de finalización fijados por los ponentes. Se ha pedido a la Escuela Superior de Informática que planifique las *N* conferencias distribuyéndolas entre las distintas salas disponibles, de forma que no haya dos conferencias en una sala al mismo tiempo. **El objetivo es minimizar el número de salas utilizadas**, para así causar el menor trastorno al resto de las actividades académicas. Construya un **algoritmo voraz** que resuelva el problema y calcule la complejidad del algoritmo desarrollado.

33.– La jefatura de estudios de la Escuela Superior de Informática cada curso académico nos plantea el siguiente problema: una vez fijados los horarios de las *N* clases a impartir (una clase es una terna <*día_semana*, *hora_inicio*, *duración*>) debe asignarse un aula a cada clase. El único requisito para que la asignación sea válida es que no puede haber dos clases el mismo día y a la misma hora que compartan aula. Se considera que ninguna clase acaba en un día diferente al día en que se inicia.

Diseñe un algoritmo eficiente que encuentre una asignación válida de aulas a las clases de modo que se emplee el **mínimo número de aulas** posible. La complejidad en espacio de este algoritmo debe ser lineal. Calcule la complejidad del algoritmo desarrollado.

34.– Juan Gorrilla, el aparcacoches, quiere trabajar menos y cobrar más (como cualquier hijo de vecino). Su trabajo le da la oportunidad de ganar mucho dinero gracias a las propinas que le dan sus “generosos” clientes. El problema es que está harto de tener que mover siempre los coches para poder conseguir aparcar un coche más (y por tanto, ganar una propina más). Juan tiene a su cargo una calle completa (con dos aceras) de *L* metros para poder aparcar los coches y cuenta con un conjunto de *N* coches a aparcar.

Diseñe un **Algoritmo Voraz** que ayude a Juan a encontrar la **mejor disposición de los coches en las dos aceras de la calle**, sabiendo que cada coche tiene una longitud diferente y cada dueño de coche da una propina distinta, de manera que **Juan gane la mayor cantidad de dinero con las propinas aunque no consiga aparcar todos los coches**. Calcule la complejidad del algoritmo desarrollado.

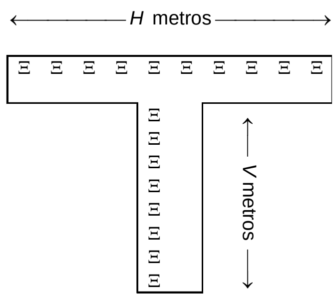
¿Es óptima la estrategia desarrollada? En caso contrario exponga un contraejemplo.

NOTA: Considere que en la longitud de cada vehículo, ya está incluida la distancia necesaria para poder aparcar el coche.

35.– En esta ocasión, Juan, *"el gorrilla"*, tiene que trabajar en una exposición de coches. Para los organizadores, es crucial que todos los coches que se aparquen de la misma marca (no es necesario aparcarlos todos) tienen que estar en la misma acera (aunque no es necesario que estén aparcados consecutivos). Los organizadores saben, a priori, que tendrán *M* marcas de coches distintas para aparcar en las dos aceras.

Resuelva el problema anterior utilizando una ***Heurística Voraz*** para esta nueva situación que se le ha presentado al *Gorrilla*.

36.– Juan, "*el gorrilla*", ha sido contratado por el Casino DSAS3, sito en el número 13 de la Calle del Percebe (en donde, hasta hace poco, se erigía un antiguo bloque de pisos, habitado por excéntricos inquilinos). Su nuevo trabajo consiste en aparcar los coches de los clientes en el patio interior del inmueble. Dicho patio, reconvertido en parking de lujo, tiene forma de "T", de forma que se pueden aparcar coches en batería en el lado horizontal de dicha "T" y en línea o cordón en el lado vertical de la "T", que sirve además como pasillo de entrada al patio.



Supuesto que Juan recibe una propina dada por cada coche que aparca, desarrolle un algoritmo basado en una ***Heurística Voraz*** que ayude a Juan a obtener la mayor suma de dinero en propinas, en los dos casos siguientes: i) Todos los coches dan la misma propina, y ii) No todas las propinas son iguales.

Nota 1: En el patio solo cabe una hilera de coches en batería y otra a uno de los lados del pasillo de entrada.

Nota 2: Cada coche que llega al casino tiene unas dimensiones específicas A_i x L_i, que ya incluyen el espacio de separación necesario para poder aparcarlo.

37.– Resulta que “*el gorrilla*” se ha vuelto un poco maniático con los coches de los colores rojo, amarillo, verde y azul. Y sólo con los coches de estos colores. Así, solo piensa aparcar los azules y verdes en batería, y los rojos y amarillos en cordón.

Resuelva el problema anterior utilizando una ***Heurística Voraz*** teniendo en cuenta esta nueva circunstancia.

38.– Se va a celebrar un maratón de cine en el que se van a proyectar durante todo un día películas, todas diferentes, en las *N* salas disponibles. Cada película se proyecta una única vez, y de ella se conoce la duración, la hora de comienzo y la sala en la que se proyectará. Nuestro viejo conocido, Juan, “*el gorrilla*”, que es un cinéfilo empedernido. Harto de tanto mundial, con las pelas que se ha sacado de aparcacoches se ha comprado un bono que le permite ver todas las películas que desee. Y piensa sacarle el máximo partido al bono **viendo el mayor número de películas posibles**.

Diseñe un ***Algoritmo Voraz*** que ayude a Juan “*el gorrilla*” a conseguir su **objetivo** (<u>**ver el máximo número posible de películas**</u>). ¿Proporciona el algoritmo una solución óptima? Calcule la complejidad del mismo.

39.– Disponemos de *n* cajas distintas, cada una de las cuales tiene un peso P_i y puede soportar una carga encima de ella C_i. Diseñe una **heurística voraz** que sea capaz de encontrar la disposición de un subconjunto de las *n* cajas de manera que se apile el mayor número de cajas. Calcule la complejidad del algoritmo.

40.– Manolo y Benito necesitan hacer una “ñiapa” en el tejado del *Empire State*, y para ello nada mejor que montar un andamio desde el piso bajo hasta el tejado, ¿verdad?

En la furgoneta llevan un numero limitado de piezas para montar los andamios hasta llegar a esa altura, pero Benito, en su línea de seguir la *Ley del Mínimo Esfuerzo* quiere hacerlo con las mínimas piezas posibles. Conocidos para cada pieza el Peso que Soporta, el Peso de la pieza y su Altura, realice un programa usando un **Algoritmo Voraz** que le diga a Benito qué piezas usar para **montar el andamio de forma que aguante el peso de ambos, se use el menor número de piezas posibles y se llegue hasta el tejado del Empire State**.

41.– El encargado de una red de concesionarios ha decidido rebajar el precio de una serie de *N* coches que no han sido vendidos y que se han pasado de moda. Los coches son voluminosos, con una longitud y un peso determinados. Para la distribución de los coches se utilizan *M trailers*. Cada trailer tiene una longitud dada, y puede soportar un peso determinado (no tienen porqué ser los mismos para todos). <u>Se deben repartir la mayor cantidad posible de coches</u>. Diseñe una **heurística voraz** para resolver el problema. Calcule la complejidad del algoritmo desarrollado.

42.– Disponemos de n cajas de longitud C_i y anchura M cada una. Se pretende empaquetar K objetos, de longitud O_j ($O_j > M$) y anchura M cada uno, utilizando el mínimo número de cajas. Diseñe un algoritmo, utilizando una **heurística voraz**, que nos de la mejor disposición de los objetos en las cajas para conseguir el objetivo de utilizar el menor número de cajas. ¿Es óptimo? Calcule su complejidad.

43.– Debido a una avería en el Sistema Informático de Control del Tráfico Aéreo, los pilotos de los aviones no saben qué ruta es la adecuada para su vuelo. Afortunadamente *Alavi Oncito* es el controlador al mando. No pudo entrar en la Escuela de Pilotos por su reducida estatura, tuvo un desarrollo tardío, aunque después creció tanto que al encargado de las pruebas le sacaría la cabeza y parte del cuello (en sueños, no figuradamente). El caso es que los N aviones tienen un destino común para todos, pero hay diferentes rutas, y cada avión sólo puede volar en un rango de alturas para hacerlo con seguridad. Disponemos pues de M rutas al destino, que están determinadas para hacerse a una altura determinada, y en las que, por razones de seguridad, no podrá haber más de un avión. Diseñe un ***algoritmo voraz*** para evitar el mayor número de problemas en los vuelos.

44.– En la ciudad ha habido una invasión de *vampiros*, motivo por el cuál *La Autoridad* ha acordado permitir que en el jardín de cada casa (e incluso en los edificios de la Universidad) se puedan plantar ajos para espantar a los "*malignos*".

Así, en la reciente construcción de una nueva urbanización, financiada por el ayuntamiento, se ha tenido en cuenta plantar en la parcela de cada casa ajos que protegerán a sus habitantes del ataque de los vampiros. Aunque tal medida tiene como contrapartida un determinado olor a ajo en cada casa.

Las casas que cuentan con financiación del ayuntamiento deben ser asignadas a las familias que las han solicitado. Para evitar quejas (e incluso abandono de la casas de la familia a la que se ha asignado) **se debe cumplir que el olor a ajo de la casa debe ser soportado por la familia que la habite**. Para cada familia tenemos el grado de tolerancia al olor a ajo.

Se trata de **maximizar el número de casas asignadas** a las familias, con la restricción anterior. Resuelva el problema de forma eficiente en tiempo y espacio. **Calcule la complejidad del algoritmo desarrollado**. **NOTA**: Cada casa se asigna a una única familia.

45.– **Viernes13.com**. *Jason* había decidido cambiar sus hábitos. Últimamente le habían estado comiendo el tarro con las bondades de la aplicación de la informática a problemas de optimización, y él que siempre estaba pensando en lo mismo, pensaba ir ahora por los cibercafés y otros garitos.com.

Para ello, se había provisto de un plano en el que se detallaba la distancia (en tiempo) t_{ij} entre los garitos.com, y también tenía una estimación del número de posibles víctimas de cada garito.com para esta noche v_i, porque *Jason* sólo tenía esta noche para cometer sus fechorías. Y también sabe el tiempo que tardará en "liquidar" a todas las víctimas de cada garito t_i (distinto para cada garito).

Había pensado en un algoritmo de rápida ejecución en encontrar la secuencia de garitos.como a visitar. Esto es, una **heurística voraz**, tratando siempre de “cazar” la mayor cantidad de víctimas a lo largo de toda la noche. Programe un algoritmo que nos devuelva la secuencia de garitos.com que visitará *Jason*, suponiendo que puede empezar por uno cualquiera de los garitos.com. Razone la optimalidad del algoritmo diseñado, y en caso de que no lo sea, exponga un contraejemplo.

46.– Con motivo de las fiestas de san Fermín, se ha ideado una campaña de promoción de los vinos de la tierra, que consiste en que cada taberna ofrece una serie de “paquetes” de chupitos. Cada paquete tiene una serie de chupitos distintos, y cada chupito tiene una graduación alcohólica dada. Un mismo tipo de vino puede aparecer en distintos paquetes, y lo mismo en distintas tabernas.

Beb Edor y su amigo *Bor Achon*, acaban de llegar a Pamplona para correr los encierros, y no piensan desaprovechar la oportunidad de probar cuantos más caldos distintos mejor. Ahora bien, puesto que piensan correr los sanfermines no sobrepasarán una graduación alcohólica total de *G* grados. Además, para quedar bien en la taberna, si se piden un paquete de chupitos, se lo beberán entero.

Puesto que ambos no están para cálculos complicados, piensan seguir una **estrategia voraz** para lograr su objetivo de beber la mayor cantidad de chupitos distintos sin sobrepasar la graduación alcohólica *G aprovechando las ofertas de las tabernas*, como buenos amigos que son se van a pedir los mismos paquetes.

47.– Se tiene un conjunto de palabras que se quieren agrupar según su grado de semejanza. Desarrolle una **heurística voraz** de manera que las palabras queden agrupadas hasta un umbral dado δ, sabiendo que al unir dos grupos, la semejanza del nuevo grupo con todos los demás es el mínimo de los dos valores de semejanza.

La **semejanza** o proximidad de dos palabras $a = a_1a_2 \ldots a_m$ y $b = b_1b_2 \ldots b_n$ está dada por:

$$\frac{2Card(A \cap B)}{Card(A) + Card(B)}, \text{ siendo } A = \{ a_ia_{i+1} / 1 \le i \le m\text{-}1\} \text{ y } B = \{ b_ib_{i+1} / 1 \le i \le n\text{-}1\}$$

Nótese que *A* y *B* son conjuntos, y por tanto, no tienen elementos repetidos.

48.– Un destacamento del ejército español con *M* hombres se ha quedado aislado en el desierto de Irak, cerca de la ciudad de Kirküt.

Han conseguido enviar un aviso por radio a su base de operaciones, pero a los servicios de salvamento les es imposible ir a rescatarles, de momento. Y han estimado que tardarán *N* días en ir a por ellos.

Tras muchas negociaciones han conseguido comprar a las guerrillas para que les suministren agua para su aseo personal.

a) Saben cuanta agua les suministraran cada día (d_i).

b) Saben el agua que necesita cada soldado para su aseo personal (a_i).

Quieren diseñar un **algoritmo voraz** que les permita mandar a los soldados un cuadrante de los días que se podrán duchar, imponiendo que sus muchachos se duchen al menos una vez y que ninguno podrá ducharse dos veces el mismo día, hasta que consigan rescatarlos.

49.– **Red de Espías**. Los servicios de Contraespionaje han detectado que una potencia "*amiga*" ha creado una tupida red de espionaje en el aparato de Seguridad del Estado. Cada espía tiene una serie de contactos con otros espías. *SuperBond*, ante tales revelaciones, se ha mostrado muy contrariado ante sus colaboradores por un aparato de seguridad trufado, que parece un *QG*, un *queso gruyère*.

En un primer momento *SuperBond* había pensado en encargarle a la *TIA* la desarticulación de la red de espías, pero debido a que Mortadelo y Filemón se encuentran en la difícil misión de proteger al *SuperJefe* en su primer encuentro al más alto nivel con precisamente dicha potencia "*amiga*", se ha decidido por encargar la misión de neutralización al *Servicio de Inteligencia Militar e Investigaciones Off* (*SIMIO*, también conocido como *SIMeIO*), que últimamente debido al escaso presupuesto, y por motivos promocionales, está colaborando con voluntarios del *Servicio Social*; habiendo creado una unidad al efecto, la *SIMSESO*.

Aunque dispuesto *SuperBond* a que no se produzcan dispendios, les ha planteado que le den soluciones a diversas alternativas, porque ya ha tenido realizar cierta "*inversión*" en comprar la información (a otra potencia "amiga" de ambas) relativa a la estructura de la red. Esto es, los contactos que mantienen entre sí los espías, y su lugar de trabajo, y otros asuntos. Pero por más que ha insistido no ha podido "sacarles" la identidad de los *topos*. (NOTA: Asumimos que un espía aislado no genera ningún peligro puesto que no puede transmitir información.)

a) El primer supuesto es que *SuperBond* no está dispuesto más que a liberar recursos para neutralizar *M* espías. Así, se trata de decidir cuáles son los *M* espías a neutralizar para causar el mayor daño a la red, esto es, romper el mayor número de contactos.

b) El segundo supuesto es cuantos espías deberían neutralizar para destruir completamente la red de espionaje.

Resuélvase cada uno de los apartados utilizando una ***Heurística Voraz***. Debe calcularse la complejidad del algoritmo desarrollado (basta con uno de los apartados).

50.– En una clase de primaria el maestro se da cuenta de que dejando a los alumnos sentarse como quieran, la clase se le "sube a las barbas". Para evitarlo ha decidido hacer él la asignación de los alumnos a los pupitres. En cada pupitre se pueden sentar dos alumnos como máximo. Hay un total de *T* pupitres y *A* alumnos. Con la observación del comportamiento de los alumnos, el profesor ha completado una tabla, *C*, del efecto que cada alumno produce sobre cada uno de los demás cuando se sientan juntos. El efecto puede ser un número positivo (le ayuda con los ejercicios, no habla con él, ...) o negativo (le mete cizaña, habla con él, le quita la goma, ...). Para algunos alumnos no ha podido comprobar el efecto de que estén sentados juntos. Si no se tiene información acerca de poner juntos *a* y *b*, tenemos $C[a, b] = 0$.

El objetivo es hacer una asignación de los alumnos a los pupitres (a cada pupitre se asignará cero, uno o dos alumnos) de una manera rápida, intentando **maximizar el efecto positivo**. Diseñe un algoritmo basado en la técnica voraz para resolver el problema. El algoritmo diseñado, ¿garantiza la obtención de la asignación óptima? Calcule la complejidad del algoritmo desarrollado.

51.– El *Jaro* "*Mantalombro*" es el alumno más "*gorrón*" de la Universidad. Su dedicación durante todo el día es estar tumbado en el sofá jugando a las consolas. Recientemente ha adquirido una consola "*Polystation*" de última generación, ya que había conseguido pasarse todos los juegos de las otras consolas que poseía. Hasta aquí todo bien, pero no había caído en un pequeño detalle y es que no poseía ningún juego para esta nueva consola y no está dispuesto a perder ni un solo minuto, ya que su cerebro le solicita jugar en todo momento. Su idea es la siguiente: sus "*contactos*" han localizado a N posibles "*pardillos*" de los cuales puede aprovecharse para grabar juegos compatibles con la "*polystation*". Cada pardillo p de estos posee m_p juegos y los contactos estiman un tiempo t_i en grabarlos. Plantee una ***estrategia voraz*** que le permita al "*Jaro*" grabarse la mayor cantidad de juegos distintos suponiendo que tiene un tiempo *máximo* T.

52.– José Peco Pión, piensa aprobar la asignatura sin más esfuerzo que el necesario para "*trasladar*" las ideas de otros a su examen. Para ello, en vez de estudiar ha ideado un procedimiento de "*traslación*" de respuestas de los problemas del examen, estableciendo previamente acuerdos con los posibles sujetos que pueden resolver los problemas ("empollones", en lo que sigue), y que también concurren al examen, que a cambio de una módica cantidad dineraria (dependiente de cada empollón), le entregarán uno de los problemas resueltos correctamente.

El procedimiento ideado consiste en hacer "*transacciones*" con empollones distantes al menos *DMin* metros (los "vigilantes" han tomado nota de las posiciones tomadas por los concurrentes al examen, y concienzudamente comprueban todos los exámenes que distan menos de la citada distancia *Dmin*) y menos de *DMax* metros (su movimiento sería detectado por algún vigilante). La transacción se reducirá a un solo ejercicio por empollón, para evitar problemas derivados de la "similitud" de los exámenes, y que se destape el "asunto".

En el examen se han propuesto N problemas, de los que se tienen que **resolver *M* correctamente para aprobar**. Cada "empollón" sólo está dispuesto a intercambiar unos cuantos problemas de entre los N, no necesariamente todos.

El tiempo que tarda en realizarse cada "intercambio" $T_{Intercambio}$ puede considerarse constante, y es muy inferior, aunque no debe ignorarse, con respecto al tiempo que se necesita para "trasladar" la respuesta del empollón, al examen de Peco (tiempo que tarda en copiarlo Peco), que está dado por t_{pe}, para el problema p resuelto por el empollón e. El importe cobrado por un empollón e es el mismo para cualquiera de los problemas que está dispuesto a intercambiar, C_e.

Resuelva el problema utilizando una estrategia **voraz**, supuesto que dispone de un tiempo T para realizar el examen (se conforma con aprobar) y desea gastar la menor cantidad de dinero. Calcule la complejidad del algoritmo desarrollado.

53.– Darío, "*el Presi*", después de fallar por sexagésima-sexta vez en su intento de fuga, ha sido obligado a trabajar como cartero dentro de la prisión. Su trabajo consiste en repartir la correspondencia a todos los reclusos de la prisión. Para Darío, esto es una contrariedad, ya que no le permite concentrarse en su *hobby* preferido, planear una nueva fuga. Por esto, ha decidido realizar su trabajo de forma correcta (para evitar represalias por parte del alcaide de la prisión), pero empleando el menor tiempo posible. Para planificar el reparto de hoy, Darío ha estado midiendo la distancia existente entre los distintos puntos de entrega de la correspondencia, y ha construido una matriz de distancias, con el tiempo de desplazamiento, y el tiempo empleado en entregar la correspondencia en cada punto. Ayude a Darío a construir un algoritmo, basado en una ***Heurística Voraz***, que le permita repartir toda la correspondencia en el menor tiempo posible, sabiendo que Darío cuenta con un carrito en el que puede llevar toda la correspondencia.

54.– Nos encontramos en época de sequía y la Junta de Regantes ha racionado el suministro de agua. Concretamente a la finca en que realizan su trabajo los reclusos, en situación de Tercer Grado, de la prisión en que se encuentra recluido Darío “*El Presi*”, le ha sido asignado un cupo de T unidades de agua. La finca está compuesta por un conjunto de P parcelas. Cada parcela p tiene una extensión e_p. En cada parcela hay sembrado un cultivo (al principio no estaban previstas restricciones de agua). De cada cultivo c se sabe tanto la necesidad de agua por hectárea, a_c, como su rendimiento neto (en euros), r_c, también por hectárea.

Lamentablemente, se sabe que si las necesidades de agua de un cultivo no son totalmente satisfechas, se perderá completamente la cosecha; no obteniéndose pues, rendimiento alguno. De la misma forma, y dado el diseño del sistema de regadíos de la finca, aunque es posible regar todas las parcelas al mismo tiempo, es imposible regar sólo una parte de una parcela, pues el agua dedicada a una parcela se distribuye uniformemente por ella.

El alcaide pretende **maximizar los beneficios que se pueden obtener de los cultivos de la finca de la prisión**, puesto que los ingresos esperados por los cultivos iban a ser empleados, a partes iguales, –según el alcaide–, en “rehabilitaciones y mejoras de las viviendas de los empleados de la prisión”, y en “mejoras de las instalaciones penitenciarias”. Todo ello, tras restar los “*gastos varios*” (que se supone que superarán el 30%, pues se rumorea incluyen un 20% para las dependencias [privadas] del alcaide, y alguna que otra comisión del 3%).

Así pues, y para motivar a los reclusos y hacerles partícipes de los éxitos de su gestión económica –y no sólo rehabilitadora–, ha convocado un **concurso entre los reclusos para determinar cuáles son las parcelas que hay que regar para maximizar la productividad (monetaria) de la finca**. El premio del concurso es una reducción de la pena a la mejor propuesta, junto con una celda individual de algo más de 25m^2 durante un año, sin compensación económica, puesto que según dice no estaría bien visto que los reclusos se lucraran gracias al sistema penitenciario, pues bien pudiera acabar derivando en un incremento de la criminalidad.

Conocidas la condiciones, Darío “*El Presi*”, recordando las conversaciones (o intercambio de informaciones) que mantenía con su compañero de celda, “*El Teclas*”, alias “*Rekcah Inverso*” en el exterior (“*El Teclas*” es un genio de la Informática con el que en tiempos pasados compartió celda, afortunadamente rehabilitado para la sociedad, y que realiza trabajos para alguna que otra empresa de seguridad. Aunque también es cierto que de vez en cuando se relaciona con “*El Puertas*”, también conocido en ciertas esferas por “*El Señor del Lado Oscuro*”). Darío, en aquellas conversaciones había adquirido algún que otro conocimiento de programación, y estaba dispuesto a desarrollar un algoritmo basado en una heurística voraz para conseguir la intimidad que proporciona una celda de 25m^2. ***Diseñe uno de los buenos***

algoritmos basado en una heurística voraz de los que podría crear Darío para resolver el problema. Calcule la complejidad del algoritmo desarrollado.

55.– *Papá Noel (Santa* para los amigos) y su banda de rufianes renos, entre los que se encuentra el malvado *Rudor* está en crisis. Los tiempos cambian y los niños de hoy en día prefieren los regalos de *Los Reyes Magos*. En una acción desesperada por conseguir el mercado de los regalos, *La banda de Santa* intentará dar el golpe de su vida robándole el mayor número de fábricas de regalos a *Los Reyes Magos*.

Disponen de N grupos de renos listos para atacar. Cada uno de estos grupos consta de R_i renos enloquecidos por la furia de la victoria. El día clave para llevar a cabo la misión "*Feliz Año Nuevo*" será el 1 de Enero. Intentarán maximizar el número de fábricas conquistadas consiguiendo así sus regalos. *Los Magos* tienen M fábricas cada una defendida por C_j camellos. Para conseguir la victoria por parte de *La banda de Santa* tendrán que atacar con mayor número de renos que de camellos defiendan la fábrica. Los grupos de renos nunca se pueden dividir.

Diseñe algoritmos basados en una estrategia ***Voraz*** que nos devuelva la asignación que **maximiza el número de fábricas** que podrán tomar *Santa* y sus renos, calculando la complejidad del algoritmo desarrollado, bajo las siguientes hipótesis:

a) Los grupos de renos no se pueden unir. Calcule la complejidad.

b) Los grupos de renos pueden unirse para asaltar una fábrica.

56.– A la peluquería "*Va yapelo's*" han llegado a la vez y sin ningún orden N clientes habituales. Como los clientes son conocidos, el dueño de la peluquería sabe el tiempo que va a tardar en atender a cada cliente (t_i), el dinero que debe cobrarle (d_i) y la paciencia (tiempo de espera) que tiene cada persona (p_i).

Diseñe un ***Algoritmo Voraz*** que ayude al peluquero a recaudar la mayor cantidad de dinero posible, aunque se vaya algún cliente sin ser atendido.

Resuelva el problema para el caso particular en el que $t_i = t, 0 \leq i < N$.

57.– Disponemos de un conjunto de N trabajos. Para cada trabajo i sabemos la fecha límite en que debería ser realizado d_i y la penalización p_i en que se incurre si no se realiza antes. Deben realizarse todos los trabajos. Los trabajos pueden procesarse en cada instante de tiempo (0, 1, 2, etc.), pero sólo uno en cada instante. **El objetivo es realizar todos los trabajos minimizando la penalización total**. Diseñe un ***Algoritmo Voraz*** que resuelva el problema. Calcule la complejidad del algoritmo desarrollado.

58.– En la empresa Losa Preta, O.S. andan escasos de espacio y necesitan distribuir los trabajadores de una forma óptima. Disponen de M puestos de trabajo y N trabajadores. Cada trabajador i tiene unas restricciones horarias r_i, una carga de trabajo c_i y una preferencia p_{ip} por cada puesto p de trabajo. Naturalmente en un puesto no pueden estar trabajando al mismo tiempo dos trabajadores distintos.

Diseñe ***heurísticas voraces*** para determinar el horario de tal forma que todos los trabajadores cubran su carga de trabajo y se **maximice la preferencia del conjunto**, según las dos alternativas siguientes:

a) Un trabajador sólo trabaja en un puesto. Esto es, si se le asigna el puesto *k*, entonces solo trabaja en el puesto *k*.

b) Un trabajador puede trabajar en varios puestos en distintas sesiones, pero no cambia en sesiones consecutivas. Por ejemplo, si de 9:00 a 10:00 trabaja en el puesto *k*, y si trabaja también de 10:00 a 11:00, entonces también lo hará en el puesto *k*.

Naturalmente puede ocurrir que las restricciones sean demasiado fuertes y el problema no tenga solución en los términos planteados. En tal caso, determine **cuál es el mínimo número de puestos de trabajo necesarios para satisfacer las restricciones horarias**, y supuesto que los empleados tienen una preferencia nula por los nuevos puestos añadidos determine la **distribución que maximice la preferencial global**.

NOTAS: La carga de trabajo es el número de horas semanales que tiene que trabajar. Las restricciones horarias se entiende que son las horas a las que puede trabajar durante la semana. La preferencia dada se entiende por unidad de tiempo.

59.– Disponemos de un conjunto de placas rectangulares de tipo *R*, cada una de dimensiones $A \times L$. Se desean recortar *n* piezas rectangulares de dimensiones $a_i \times l_i$, de tal forma que se minimice el número de las placas de tipo *R* de las que se ha obtenido alguna pieza. Diseñe una ***Heurística Voraz*** que resuelva el problema.

60.– Tenemos un procesador que debe realizar una serie de *N* trabajos. De cada trabajo *i* sabemos el instante exacto c_i en que debe comenzar, su duración d_i y el beneficio b_i que proporciona si se realiza. El procesador sólo puede realizar una tarea al mismo tiempo. Supuesto que el procesador está operativo entre los instantes *C* y *F*, diseñe un ***Algoritmo Voraz*** que maximice el beneficio de los trabajos que se procesen. Calcule la complejidad del algoritmo desarrollado. ¿Es óptimo el algoritmo desarrollado? Si no lo es, debe exponer un contraejemplo. En todo caso, debe razonarse la respuesta.

61.– Marcos es el encargado de colocar los cuadros en el Museo Artístico Nacional. El director de este museo tiene la mala costumbre de cambiar los cuadros demasiado a menudo y Marcos está harto de trabajar para colocar y recolocar los cuadros una y otra vez. El museo tiene *S* salas de dimensión $A_j \times L_j$ metros, para cada sala *j*. Cada sala puede tener hasta cuatro puertas, que dan a salas contiguas. Todas las puertas miden *P* metros. Cada cuadro tiene una anchura C_i.

Diseñe un algoritmo que ayude a Marcos a colocar los *N* cuadros de la exposición de manera que se utilice el menor número de salas posible.

Nota 1: No se puede colocar un cuadro encima de otro.

Nota 2: Considere que en el ancho de cada cuadro C_i ya está incluido el espacio de separación entre cuadro y cuadro.

62.– La empresa *Enredando Inc*. desea contar con las últimas tecnologías y ahora se ha embarcado en el proyecto de dotar a todas sus instalaciones con red inalámbrica (WiFi). La responsabilidad de la implantación de la red WiFi ha recaído en su departamento técnico, que ha realizado un estudio al respecto. En el estudio se incluye un mapa de las instalaciones de la empresa en el que se muestra la disposición de los puntos en los que se pueden instalar las antenas, los puntos de la empresa que deben tener cobertura y la distancia [lineal] existente entre cada par de puntos.

Las especificaciones técnicas de las antenas WiFi indican que tienen una cobertura circular de *R* metros de radio y pueden ser configuradas en 3 canales diferentes. Además, advierten que en el caso de que exista un solapamiento en la cobertura de las redes, este solapamiento no debe ser de señales pertenecientes al mismo canal, pues se producirá una interferencia en las señales y la zona de intersección quedará sin cobertura.

Diseñe un ***Algoritmo Voraz*** que encuentre la disposición óptima de las antenas (posición y canal), de tal manera que utilizando el menor número de antenas se proporcione cobertura a todos los puntos especificados, evitando las interferencias. El problema planteado puede no tener solución, exponga las alternativas para conseguir una solución al problema, teniendo presente que el principal objetivo es conseguir la cobertura total.

63.– La empresa de instalaciones eléctricas "*Los Bombillas*", ha resultado la adjudicataria del concurso para la instalación del alumbrado de la ciudad. Han realizado un estudio y disponen de un plano en el que se detallan los puntos que deben iluminarse, y que son los mismos en los que es posible instalar farolas, además tenemos la distancia lineal entre ellos, y se sabe que cada farola tiene un radio de iluminación de *R* metros. Se trata de **iluminar todos los puntos señalados minimizando el número de farolas empleadas**. Diseñe un ***Algoritmo Voraz*** que resuelva el problema.

64.– El padre *Donateo* ha pronosticado el fin del mundo y todos en el pueblo de *Melocreoto* están un poco desconcertados. Tal ha sido el revuelo, que un pastor afincado en esta localidad ha decidido crear un arca bendecida por el cura para así intentar salvar a sus ovejas. El pastor, que es buena gente, además de a sus ovejas ha decidido meter en su arca a animales a los que les tiene especial aprecio. Como a su burro Rucio, al lagarto del Tomás, a la gata de su hija, a su suegra, a un loro, a la cotorra de su cuñada, al toro de grandes cuernos del alcalde… y a otros cuantos animales más de distintas especies de los que meterá una pareja.

En una cuadra sólo meterá a animales de una misma especie (la cotorra de su cuñada y la gata de su hija no se llevan muy bien). Para cada pareja animal y animal individual, se saben su peso y las dimensiones mínimas del compartimento a ocupar (se incluyen las necesidades de pienso que necesitan). Además, el total de animales no debe de hundir el maravilloso arca y que mueran todos ahogados.

Diseñe un ***Algoritmo Voraz*** que ayude al pastor a meter a la mayor cantidad de animales posibles. Calcule la complejidad del algoritmo desarrollado.

65.– En el *Mundo de Oz* están pavimentando los caminos que unen las distintas ciudades con baldosas amarillas (todas de igual tamaño). El *Mago de Oz* se ha encontrado con un problema económico, pues su mundo es más grande de lo que pensaba, y hay muchos caminos que embaldosar. Dado que sus arcas no están rebosantes de dinero, el *Mago* pretende embaldosar los caminos que comunican todas las ciudades del *Mundo de Oz* con la *Ciudad Esmeralda*, la gran capital del imperio, pero **utilizando el menor número de baldosas amarillas**.

Diseñe un ***Algoritmo Voraz*** que dado el mapa de caminos del *Mundo de Oz*, resuelva el problema que tiene el *Mago de Oz*. Calcule la complejidad del algoritmo desarrollado.

66.– En una ciudad el alcalde ha decidido asfaltar sus calles por motivos electorales, pero tiene el problema de que no puede asfaltar todas sus calles por motivos de presupuesto. Ha decidido asfaltar aquellas calles que unan una plaza con otra, quedando todas las calles asfaltadas unidas. Para cada calle y cada plaza sabe cuánto cuesta asfaltarla, y también su tiempo en hacerlo. Además, sabe por estadísticas de elecciones anteriores y encuestas realizadas recientemente, el número de potenciales votantes de su partido tanto para cada calle, como para cada plaza.

El alcalde se ha propuesto maximizar el número de sus partidarios con plaza o calle asfaltadas, puesto que cree que contribuirá a conseguir afianzar el voto entre los seguidores acérrimos, y en el resto, a inclinar definitivamente la balanza a su favor. Diseñe una ***Heurística Voraz*** que indique cuáles son las calles y plazas a asfaltar, supuesto que disponemos de un presupuesto P.

67.– *De Campaña*. Tenemos un mapa de carreteras con N puntos críticos, que podemos clasificar en *Puntos Negros* y/o *Recaudatorios*. Dado un periodo determinado de tiempo fijo, para cada punto sabemos la recaudación que se obtiene y el número de accidentes que se producen atendiendo a estas tres alternativas: (1) no cubrirlo; (2) cubrirlo con un radar; (3) cubrirlo con una dotación de agentes de tráfico. Disponemos de R radares y P dotaciones de agentes de tráfico ($P < R$). Se pide resolver los siguientes problemas:

a) Maximizar la recaudación sin que se sobrepase una cantidad M de accidentes prefijada de antemano.
b) Minimizar el número de accidentes.

Diseñe ***Algoritmos Voraces*** que resuelvan los problemas planteados.

68.– La Prima Vera ha venido a pasar un tiempo con nosotros, y para no variar, nos ha traído como presente un buen ramo de N flores. Las flores son voluminosas, pero de lo más hermosas, y cada una tiene un peso y volumen determinados. Como somos previsores hemos colocado por toda nuestra casa un conjunto de M jarrones para colocar la mayor cantidad de flores y complacer lo más posible a nuestra querida prima. Utilizando ***Algoritmos Voraces*** resuelva cada uno de los problemas siguientes:

i) sólo colocamos una flor en cada jarrón,
ii) podemos colocar más de una flor por jarrón.

69.– Juanito se ha comprado una cosechadora para vendimiar. Se aproxima la época de la recolección de la misma y a Juanito le llegan muchos clientes para pedirle que recoja su cultivo ya que éste tiene unos precios muy competitivos. Ante la avalancha de clientes, Juanito tiene que hacer una selección de los mismos ya que le va a ser imposible atender a todos porque la uva tiene una fecha a partir de la cual pierde su valor. De cada parcela se sabe precio que tiene que cobrar, el tiempo que emplea en la recolección y la fecha a partir de la cual no merece la pena recoger el producto porque “caduca” y no se producen beneficios.

Diseñe un ***Algoritmo Voraz*** que ayude a Juanito a maximizar el beneficio puesto que necesita amortizar cuanto antes el coste de la cosechadora.

Resuelva el problema suponiendo que un cliente puede tener varias parcelas, y que todas las parcelas del cliente tienen que recolectarse consecutivamente.

70.– **La granja de Pepito**. Una comarca tiene N_G de parcelas para pastar el ganado, N_C parcelas de cultivo y N_A parcelas de arboleda. Al final de la temporada cada una de ellas aporta un beneficio B_G, B_C y B_A, respectivamente. Así, si un granjero tiene G parcelas de pastos, C parcelas para cultivar y A parcelas de arboleda, el beneficio total al final de la temporada viene dado por:

$$Beneficio = G*B_G + C*B_C + A*B_A$$

Al comienzo de cada temporada, el granjero puede decidir comprar una (y sólo una) nueva zona con sus ahorros, teniendo cada tipo de zona un precio diferente: P_G, P_C y P_A.

Además, tenemos que:

a) cada parcela de ganado nueva supone inutilizar media parcela cultivable para dar de comer a los animales,

b) cada parcela cultivable adicional supone inutilizar media de arboleda, y

c) cada parcela de arboleda supone inutilizar media parcela de ganado para su plantación.

Dado un presupuesto inicial P, unas zonas iniciales de cada tipo G_0, C_0 y A_0, se trata de **maximizar el beneficio que podemos obtener al cabo de *N* temporadas**. Resuelva el problema mediante un ***algoritmo voraz***.

71.– Tenemos la hora de paso de los autobuses de una línea metropolitana por una parada dada recogidos durante un mes. Los autobuses pasan periódicamente por dicha parada, aunque la hora de paso suele variar por las condiciones del tráfico. Supuesto que se sabe el número de autobuses que pasan por dicha parada, diseñe una ***Heurística Voraz*** que determine el horario de los autobuses en dicha parada, esto es, las horas a las que los autobuses pasan por dicha parada.

72.– Proponga un problema que pueda ser resuelto por un Algoritmo Voraz e impleméntelo.