TEORÍA DE ALGORITMOS 1

Guia de ejercicios: greedy

Por: ING. VÍCTOR DANIEL PODBEREZSKI
vpodberezski@fi.uba.ar

Enunciados

Para cada ejercicio considere, además de resolver lo solicitado, calcular la complejidad temporal, espacial, justificar la optimalidad y su pertenencia a la metodología greedy. En muchas ocasiones existen varias soluciones alternativas óptimas greedy. Tómese un tiempo para buscarlas.

Referencias

- ⭐ Fácil
- ⭐⭐ Medio
- ⭐⭐⭐ Difícil

1. ⭐Una ruta tiene un conjunto de bifurcaciones para acceder a diferentes pueblos. El listado (ordenado por nombre del pueblo) contiene el número de kilómetros donde está ubicada cada una. Se desea ubicar la menor cantidad de patrullas policiales (en las bifurcaciones) de tal forma que no haya bifurcaciones con vigilancia a más de 50 km. Proponer un algoritmo que lo resuelva.

   Ejemplo (ciudad,Bifurcación): (Castelli, 185), (Gral Guido, 249), (Lezama 156), (Maipu, 270), (Sevigne, 194). Si incluimos un patrullero en la bifurcación de Lezama, cubre además de esta a Castelli y Sevigne. Pero no Gral Guido y Maipú. Se necesitaría en ese caso, ubicar otro. Al agregar otro patrullero en Gral Guido, se cubren todas las ciudades restantes. Con 2 móviles policiales en bifurcaciones se cubren todas los accesos a todas las ciudades con distancia menor a 50km.

2. ⭐Conocemos al algoritmo de Kruskal y Prim sobre un grafo conexo y ponderado para obtener su árbol recubridor mínimo. Analice la siguiente estrategia de resolución y determine si corresponde a un algoritmo óptimo. Si lo es, detalle con qué estructuras lo implementaría de la forma más eficiente posible.

   Iniciar con el grafo completo
   Mientras existan ciclos en el grafo

Obtener la arista de mayor peso cuya remoción mantenga la conectividad del grafo
Eliminar la arista seleccionada.

3. ⭐ Sea C el conjunto de ejes seleccionados por el algoritmo de Dijkstra para un grafo G=(V,E) pesado no dirigido. Probar o dar un contraejemplo las siguientes afirmaciones:
    a. C corresponde a un árbol recubridor
    b. C corresponde a un árbol recubridor mínimo.

4. ⭐ ⭐ Para la realización del próximo congreso de "charlas motivacionales para el joven de hoy" se contrató un hotel que cuenta con "m" salas de exposición. Existirán "n" oradores. Cada uno solicitó un tiempo de exposición definido por un horario de ingreso y una duración. Los organizadores quieren asignar las salas con un intervalo entre charla y charla de 15 minutos y desean utilizar la menor cantidad de salas posibles. Presentar un algoritmo greedy que resuelve el problema indicando la cantidad de salas a utilizar y la asignación de las charlas. En caso de sobrepasar el máximo de salas disponibles informar. Analice complejidad y optimalidad

5. ⭐ ⭐ Una carrera tipo "Ironman" es un triatlón compuesto por 3 instancias: natación (3,86 km de natación), ciclismo (180 km) y carrera a pie (42,2km). Para conocer al ganador se suman los tiempos realizados en cada una de las etapas. Tanto el ciclismo como la carrera a pie se puede realizar en simultáneo con todos los inscriptos. Pero, por una regulación se prohibió que más de 1 persona realice la etapa de nado en el lago en simultáneo. Se conoce el tiempo estimado de cada participante para cada evento. Proponga un orden de salida de tal forma de minimizar el tiempo total de toda la competencia.

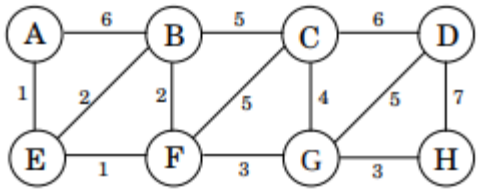
6. ⭐ Las membresías al club de vino "Varietales de cuyo" son de 4 categorías: "Titanio", "Oro", "Plata" y "Básico". Cada socio tiene una membresía. Por mes envían un pack de degustación que llaman "alfa", "beta", "gamma" y "epsilon". Un socio "Titanio" sólo puede recibir un pack "alfa", Un socio "Oro" puede recibir un pack "alfa" o "beta", un socio "Plata" sólo puede recibir "alfa" o "gamma". Finalmente un "Básico" puede recibir cualquier pack. Diseñar una estrategia greedy para resolver el siguiente problema: Sean Pa, Pb, Pg, Pe los packs disponibles de cada tipo y St, So, Sp, Sb los socios de cada categoría. Informar cómo se puede satisfacer la distribución de packs entre socios (o si no se puede satisfacer).

7. ⭐ El club de amigos de la república Antillense prepara un ágape en sus instalaciones en la que desea invitar a la máxima cantidad de sus "n" socios. Sin embargo por protocolo cada persona invitada debe cumplir un requisito: Sólo puede asistir si conoce a al menos otras 4 personas invitadas. Nos solicita seleccionar el mayor número posible de invitados. Proponga una estrategia greedy óptima para resolver el problema.

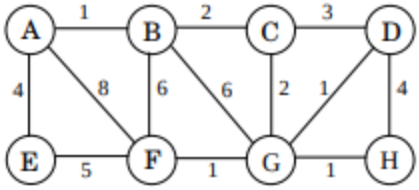
8. ⭐ ⭐ Considere el problema anterior, pero con la adición de una nueva restricción: Los organizadores desean que cada invitado pueda conocer nuevas personas. Por lo que nos solicitan que sólo puede asistir si NO conoce al menos otras 4 personas invitadas. Modifique su propuesta para satisfacer esta nueva solución.

9. ⭐ Indicar si las siguientes afirmaciones son verdaderas o falsas. Justificar la respuesta.

a. El algoritmo de Prim genera el mismo resultado que el algoritmo de Kruskal, al calcular el MST de un Grafo.
b. El algoritmo de Prim aplicado a un Grafo genera siempre el mismo MST
c. El algoritmo de Kruskal aplicado a un Grafo genera siempre el mismo MST
d. Si un Grafo posee todas sus aristas distintas, el MST es siempre el mismo sin importar qué algoritmo se utilice
e. El MST del siguiente Grafo posee 2 MST



f. El MST del siguiente Grafo es: {AB, AE, BC, CG, GD, GF, GH}



g. Dado un grafo G no dirigido con costos en cada arista c(e). Si suponemos que e∗ es la arista de menor costo (c(e∗) < c(e) para cada arista de G / e <> e∗), entonces podemos afirmar que cualquier MST de G contendrá a e*.

10. ⭐⭐Un fabricante de perfumes está intentando crear una nueva fragancia. Y desea que la misma sea del menor costo posible. El perfumista le indicó un listado de ingredientes. Por cada uno de ellos determinó una cantidad mínima (puede ser cero) y una máxima que debe contar en la fórmula final. Cada ingrediente tiene asociado un costo por milímetros cúbicos. Sabiendo que en la presentación final es de X milímetros cúbicos. Presentar una solución utilizando metodología greedy que resuelva el problema.

11. ⭐El ajedrez se juega con un tablero cuadriculado. La pieza llamada “Rey” puede moverse en cualquiera de los 8 cuadrados aledañas a su posición actual comiendo cualquier otra pieza que esté en ellos. Contamos con un tablero especial de nxm cuadrados y una cantidad ilimitada de piezas “Rey”. Queremos ubicar la mayor cantidad de reyes sin que estos se puedan comer entre si. Proponer un algoritmo greedy para resolverlo. Brindar complejidad. Justificar la optimalidad de su propuesta.

12. ⭐Un importante club de campo debe cerrar durante un mes sus instalaciones. En ese tiempo tiene que licenciar a la mayoría de sus empleados. Entre ellos a los guardias de seguridad. Cada uno de ellos trabaja 1 vez por mes en un horario de corrido por mes iniciando un día determinado y finalizando unos días después. Habitualmente varios guardias se superponen en algunos momentos del mes. Pero sabemos que siempre hay al menos 1 de ellos en las instalaciones. Los que nos solicitan es que dejemos a la mínima cantidad de guardias posibles y que aun siempre haya al menos uno custodiando durante el

mes. Proponga un algoritmo greedy eficiente que lo resuelve.

13. ⭐ ⭐ Un centro de distribución de repuestos ferroviarios se encuentra en un punto de la red de este transporte. Es la encargada de distribuir a demanda los materiales y recursos para las reparaciones que solicitan las diferentes estaciones. Como la red es antigua y está mal mantenida la cantidad de kilos que se puede transferir sobre cada trayecto es variable. Esto para ellos es un problema porque quieren enviar la mayor cantidad posible de material por viaje. Tanto es así que no les importa realizar un camino más largo siempre que eso implique transportar más materiales. Se pueden armar diferentes caminos que unan el centro de distribución con cada estación. Estos estarán conformados por una secuencia de trayectos, cada uno con su propia limitación de kilos que soporta. Llamamos cuello de botella al valor mínimo entre ellos. Construir un algoritmo greedy que permita calcular el camino con el máximo cuello de botella entre el punto de partida y el resto de los puntos.

14. ⭐ ⭐ Una jefa de trabajos prácticos (jtp) está a cargo de un grupo de N ayudantes, cada uno de los cuales tiene que trabajar un turno completo durante la semana. Hay muchas actividades asociadas con cada turno (atender el laboratorio, dar clase de consultas, etc.) pero podemos pensar en un turno como un intervalo de tiempo contiguo. Puede haber más de un turno simultáneamente. La Jtp está tratando de construir un subconjunto de esos N ayudantes para formar un comité de supervisión, que esté en contacto con todos los ayudantes. Un comité de supervisión está en contacto con todos los ayudantes si, cualquiera sea el turno de un ayudante, hay alguien en el comité de supervisión cuyo turno se superpone, aunque sea parcialmente, con el turno de ese ayudante. Ejemplo: N=3, ayudante 1 = Lunes de 16 a 20, ayudante 2 = Lunes de 18 a 22, ayudante 3 = Lunes de 21 a 23. En este caso, la solución es {ayudante 2}. Dado un conjunto de ayudantes, diseñar un comité de supervisión lo más pequeño posible, usando una estrategia greedy.

15. ⭐ ⭐ Una familia planea un viaje. Tienen pensado salir desde su ciudad a una determinada hora y llegar lo antes que puedan a la ciudad de destino. Para hacer el viaje cuentan con diferentes opciones de recorridos. Diferentes rutas pasan por diferentes pueblos y se ramifican en otras rutas diferentes. Cada trayecto cuenta con una longitud que - teniendo en cuenta el límite de velocidad - les permite establecer el tiempo que les llevará recorrerlo. Por otro lado, conocen el tráfico por hora de cada trayecto. Esto es, el tiempo extra que le insume realizar un trayecto si inician el recorrido en una determinada hora. Construir un algoritmo greedy que determine el mejor camino a realizar en el menor tiempo posible.

16. ⭐ La república soberana de Antillense tiene su territorio en una isla paradisiaca. Los últimos años sufrió grandes problemas de destrucción de su infraestructura luego del estallido de su volcán, un terremoto, un maremoto y un ataque de mosquitos mutantes radioactivos. Entre sus planes de reconstrucción deben unir mediante carreteras sus “n” poblaciones principales. En un estudio previo conocen el costo de generar caminos desde cada uno de los poblados hacia los otros. Como no tienen fondos deben solicitar préstamos. Desean gastar el menor valor posible, construyendo la menor cantidad de rutas y al menor costo total. Se debe tener en cuenta que únicamente se puede solicitar un préstamo por año para construir una única ruta. Y que la inflación en el país hace que los costos de construcción aumenten 100% . Debe indicar que rutas construir y en qué orden para lograr minimizar el costo total. Presentar una solución greedy que solucione el problema.

17. ⭐Una empresa de telecomunicaciones ganó una licitación para construir un conjunto de “r” líneas de comunicación bidireccional para “n” ciudades. Cada línea de comunicación “l” une dos ciudades y de generarla le permite obtener una ganancia mensual Gl. Puede elegir construir cualquier línea a menos que una nueva línea conecte 2 ciudades previamente conectadas que un camino que pase por 1 o más líneas. Nos solicitan nuestro asesoramiento para determinar qué líneas construir maximizando la ganancia obtenida.

18. ⭐⭐Para habilitar la realización de un importante evento multideportivo se solicitó como precondición que durante el lapso que dura cada actividad exista junto a la misma personal médico. Conocemos para cada una de las “n” actividades a realizar, el momento de inicio y final. Como encargados de la inspección nos solicitan que programemos la menor cantidad de inspecciones posibles en las que constatamos que (al menos al momento de la inspección) se cumple la precondición. Una inspección verifica únicamente aquellos eventos que se están llevando a cabo en el momento. Ninguna actividad debe quedar sin inspeccionar. Presentar una solución greedy óptima al problema.

19. ⭐Nos proponen el siguiente juego de cartas en el que tenemos que adivinar la carta que tiene un rival. El mazo tiene 1 carta de “1 de Oro”, 2 cartas de “2 de Oro” y así hasta 9 cartas de “9 de Oro”. El rival mezcla y selecciona una carta. Mediante preguntas que solo se pueden responder por sí o por no tenemos que averiguar en la menor cantidad de consultas cual es la carta. (ejemplos: “La carta es mayor a 4?, “La carta es un “1” o un “3”, etc). Proponer un algoritmo greedy que resuelva el problema minimizando la cantidad probable de preguntas a realizar.

20. ⭐⭐⭐Sean A y B dos sets de “n” puntos en el plano p=(x,y). Un punto $a_i=(x_i,y_i)$ de A domina a un punto $b_j=(x_j,y_j)$ de B si y sólo si $x_i \geq x_j$ y $y_i \geq y_j$. Un emparejamiento (match) entre un punto a_i de A y uno b_j B es posible si a_i domina a b_j. Llamamos matching M a un conjunto de emparejamientos $\{(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)\}$ y su tamaño corresponde a k. Un matching es máximo si no existe otro posible matching con mayor cantidad de puntos. Proponga una estrategia greedy óptima que obtenga el matching máximo para cualquier sets de conjuntos A y B. Procurar realizarlo con la menor complejidad espacial y temporal posible.

21. ⭐⭐Un servidor de videojuegos se alquila por horas. El contrato dura un tiempo fijo y permite utilizar en forma exclusiva el mismo por una cantidad continua de horas una vez por semana. Por cada contrato que el dueño del servidor establece, se lleva un monto fijo de dinero. Al dueño del servidor le interesa tener la mayor cantidad de contratos posibles (sin importar la duración en horas de los mismos). El servidor funciona las 24hs. Recibe un conjunto de ofertas de contrato y debe seleccionar cuales aceptar. Cada contrato tiene un día y hora de inicio y un día y hora de fin. Durante ese lapso tendrán la exclusividad del servidor. Ese tiempo contiguo no puede durar más de 1 semana (un contrato podría pedir por ejemplo 3 días completos pero nunca superar la semana).. Y esa fecha se repite todas las semanas. Los contratos aceptados no deben superponerse. Proponer una solución greedy que solucione el problema de forma óptima. Tenga en cuenta que es posible contratos que empiecen al finalizar la semana y terminen horas después del inicio de la misma.

22. ⭐Una fotocopiadora cada mañana recibe un conjunto de pedidos de clientes. El pedido del cliente i demora ti en ejecutarse. Para una planificación dada (es decir un cierto orden de las tareas) Ci es la hora en la cual el pedido i termina de ejecutarse (por ejemplo, si el pedido j es

el primero que se ejecuta, Cj = tj; si el pedido j se ejecuta a continuación del pedido i, Cj=Ci+tj). Cada cliente tiene un peso wi que representa su importancia. Se supone que la felicidad de un cliente depende de cuán rápido le entregan el trabajo, por lo que la empresa decide minimizar el tiempo de demora ponderado = Suma (wi * Ci). Diseñar un algoritmo greedy eficiente para resolver este problema.

23. ⭐Contamos con una impresora central en un centro de cómputos del campus universitario. Entre varios departamentos y laboratorios nos solicitan al inicio de cada mes, la impresión de “n” documentos. Cada uno de ellos tiene una duración determinada y cuenta con una fecha de entrega. Si nos pasamos de esta recibimos un apercibimiento proporcional al retraso más largo del mes. Como a la impresora le falta mantenimiento queremos lograr - siempre que sea posible - tiempo entre los trabajos de impresión. Presentar un algoritmo greedy que dada la lista de tareas proponga la fechas de inicio de publicación minimizando el apercibimiento y dando tiempo entre las tareas siempre que sea posible.

24. ⭐Contamos una mazo de cartas numeradas del 1 al “n” mezcladas en un orden desconocido y boca abajo (no vemos que numero tienen). No podemos modificar ese orden ni espiarlo. Debemos iterativamente tomar la carta superior, darla vuelta y apilarla. Una carta puede formar una pila nueva o ubicarse en una existente. Para ubicarla en una existente debe ser menor a la carta superior de esa pila. Una vez ubicada la carta, no se puede mover y pasa a ser la carta superior. El objetivo es ubicar todas las cartas en la menor cantidad de pilas. Presentar un algoritmo greedy para resolver el problema.

Ejercicios resueltos

Cada ejercicio resuelto busca mostrar cómo se debe analizar, resolver y justificar la resolución del ejercicio, tanto en un trabajo práctico o en un parcial

1. ⭐Un importante museo nacional tiene “n” piezas de arte en diferentes bóvedas de seguridad (1 pieza por bóveda inicialmente). Se ha programado una exposición especial que requiere que esas n piezas se reúnan para luego ser transportadas al lugar designado. Para la movilización se ha contratado un seguro que cobra un valor cada vez que una pieza corre riesgo de robo (cuando la bóveda que la contiene se abre). Cada pieza fue tasada con un valor determinado. Cuando se realiza un traslado hay 2 bóvedas involucradas. Se abren tanto la bóveda de origen como la de destino. Por lo tanto se debe pagar al seguro los valores de los que están almacenados en ambas bóvedas. En el traslado se pueden seleccionar cualquier cantidad de las piezas que están en las bóvedas involucradas. Deseamos determinar qué traslados realizar de forma de lograr unificar en una sola bóveda todos las piezas abonando la menor cantidad de plata a la aseguradora. Proponer un algoritmo greedy que solucione el problema (y que sea óptimo).

   Ejemplo: Si tengo 4 piezas con sus respectivos valores en 4 bóvedas. A: p1=5 B: p2=4 C: p3=2 D: p4=6 Si comienzo trasladando p1 a la bóveda B. Deberé abrir la bóveda A y B. Se pagará al seguro 5+4=9 por el proceso. Si ahora se desea trasladar p3 a la bóveda B se deberá pagar 2+5+4=11. El proceso continúa hasta que todas las piezas estén unificadas.

Pseudocódigo

Python

```
Sea P la lista de las piezas

Sea heap una estructura Heap de mínimos utilizada para almacenar piezas según
su valor

Sea Pieza una estructura que representará una pieza con un valor particular

Sea Translados un vector que indica qué translados se deben realizar. El primer
elemento indica el primer translado y el último elemento, el último translado a
realizar


heap = new Heap()

for pieza in P:
       heap.insert(pieza)

while heap.size <> 1:
       pieza1 = heap.pop()
```

```
        pieza2 = heap.pop()

        nueva_pieza = new Pieza(pieza1, pieza2) # representamos el translado
                                                  entre las piezas de menor valor

        heap.insert(nueva_pieza)

        # representamos el translado de la pieza1 a la ubicación de la pieza2
        Translados.push((pieza1.piezas, pieza2.location))


# Reconstruimos la solución
for (piezas, location) in Translados:
        log --> translado "piezas" a location

return Translados
```

Explicación

Para solucionar el problema se plantea utilizar principalmente un heap de mínimos, donde se almacenará piezas y su orden se dará en función del valor de las mismas. Inicialmente se agregarán todas las piezas al heap.

En cada paso el algoritmo seleccionará las dos piezas de menor valor, creará una pieza extra que contendrá a las mismas y cuyo valor será la suma de ambas piezas. De esta manera, estaremos representando el translado de las piezas cuyo valor sea mínimo. Además se guarda una referencia al translado que se realiza, siendo en de la pieza (o conjunto de ellas) de menor valor a la bóveda donde se encuentra la pieza (o conjunto de ellas) de segundo menor valor. La ejecución se repite hasta que el heap contiene 1 pieza (todas las piezas han sido transladadas).

Por último se reconstruye la solución en base a los traslados que han sido realizados.

Análisis de complejidad

Complejidad temporal

La complejidad temporal de la solución es O(N.log(N)) ya que

- la operación de insertar N elementos en un heap es O(N.log(N))
- iterar el heap hasta que solo quede un elemento se ejecuta N veces (en cada paso una pieza es trasladada a una bóveda). En cada paso se realizan dos operaciones de pop (O(logN)) y una operación de push (O(logN)) al heap, con lo cual la complejidad de la sentencia es O(N.log(N))
- finalmente la operación de reconstrucción es de O(N) ya que N son los traslados de piezas que se realizaron

Complejidad espacial

La complejidad espacial de la solución es O(N^2) ya que

- el heap contiene N elementos, lo cual posee una complejidad de O(N)

- el vector de Traslados almacenan N elementos, pero los mismos pueden contener hasta N piezas en el peor de los casos. Por ejemplo en el último paso, donde N-1 piezas pueden movidas a la última bóveda. Por lo tanto la complejidad espacial del vector Translados es de O(N^2)

Análisis de optimalidad

Por propiedad del Heap, este siempre nos devuelve el elemento de menor valor.
En cada iteración obtenemos los 2 elementos de menor valor para sumarlos y dicha suma insertarla nuevamente en el Heap. Los mismos pueden ser 2 piezas o combinaciones de ellas en bóvedas.

Para poder demostrar este ejercicio podemos hacerlo mediante árboles de Huffman, donde el concepto y procedimiento utilizado es el mismo al planteado. Paso a paso, se seleccionan los elementos de menor valor y se crea un nuevo elemento del valor conjunto.
La elección greedy puede demostrarse ya que cada elección de elementos nos acerca en cada paso a la solución global. Por cada iteración se mueve la menor cantidad de piezas posibles. Si definimos un conjunto de bóvedas, donde todas las piezas (i) tienen el mismo valor, menos 2 que tienen un valor distinto y menor a las otras (“a” y “b”). Dichas piezas serán movidas a la misma bóveda, formando un nuevo valor “z”.
Si moviéramos cualquier otra pieza que no sea “a” y “b”, el costo total del movimiento sería Wi >= Wa, Wi >= Wb, generando un costo mayor.