Algoritmos Voraces o Greedy

Emanuel Lupi

Universidad Nacinal de Córdoba

10 de julio de 2019

Contenidos

1. Algoritmos Greedy
   - Qué son los algoritmos greedy?
2. Algunos problemas greedys conocidos
   - Activity Selection Problem
   - The Investor
   - The Agency
3. Forma de probar un algoritmo greedy
4. Un problema interesante

Definición

- Un algoritmo greedy es una estrategia de búsqueda.

Definición

- Un algoritmo greedy es una estrategia de búsqueda.
- En la cual se usa una heurística consistente en elegir la opción óptima en cada paso local.

Definición

- Un algoritmo greedy es una estrategia de búsqueda.
- En la cual se usa una heurística consistente en elegir la opción óptima en cada paso local.
- El algoritmo debe conducir a una solución óptima.

Definición

- Un algoritmo greedy es una estrategia de búsqueda.
- En la cual se usa una heurística consistente en elegir la opción óptima en cada paso local.
- El algoritmo debe conducir a una solución óptima.
- Para saber que el algoritmo conduce a una solución óptima hay que demostrarlo. Y por lo general se hace un demostración formal.

Definición

- Un algoritmo greedy es una estrategia de búsqueda.
- En la cual se usa una heurística consistente en elegir la opción óptima en cada paso local.
- El algoritmo debe conducir a una solución óptima.
- Para saber que el algoritmo conduce a una solución óptima hay que demostrarlo. Y por lo general se hace un demostración formal.
- Hay problemas greedy que son conocidos y por tanto no hay que demostrarlos. Pues ya fueron probados.

Problema de la selección de tareas

Problema de la selección de tareas

Tenemos actividades a realizar, dichas actividades tienen un principio y fin conocido. Dichas actividades se pueden superponer. El problema pide la máxima cantidad de actividades que podemos elegir sin que se superpongan.

Problema de la selección de tareas

Problema de la selección de tareas

Tenemos actividades a realizar, dichas actividades tienen un principio y fin conocido. Dichas actividades se pueden superponer. El problema pide la máxima cantidad de actividades que podemos elegir sin que se superpongan.

- Por ejemplo si tenemos tres tareas de rangos (1, 3), (2, 9) y (8, 10)..

Problema de la selección de tareas

Problema de la selección de tareas

Tenemos actividades a realizar, dichas actividades tienen un principio y fin conocido. Dichas actividades se pueden superponer. El problema pide la máxima cantidad de actividades que podemos elegir sin que se superpongan.

- Por ejemplo si tenemos tres tareas de rangos (1, 3), (2, 9) y (8, 10)..
- Respuesta es 2, la primera y la última

Solución

- La forma correcta de ordenarlas es por horario de finalización.

Solución

- La forma correcta de ordenarlas es por horario de finalización.
- Si podemos realizar la próxima tarea la realizamos, sino la ignoramos, (esto es: que si la tarea siguiente empieza después de que esta termina).

Solución

- La forma correcta de ordenarlas es por horario de finalización.
- Si podemos realizar la próxima tarea la realizamos, sino la ignoramos, (esto es: que si la tarea siguiente empieza después de que esta termina).
- De esta forma, podemos pensar en que vamos realizando las tareas teniendo en mente que terminemos lo mas antes posible para tomar otra.

Prueba

- Supongamos que el algoritmo no es óptimo.

Prueba

- Supongamos que el algoritmo no es óptimo.
- Entonces hay una solución mejor que la del algoritmo greedy.

Prueba

- Supongamos que el algoritmo no es óptimo.
- Entonces hay una solución mejor que la del algoritmo greedy.
- Llamemos G a la solución greedy y O la solución óptima.

Prueba

- Supongamos que el algoritmo no es óptimo.
- Entonces hay una solución mejor que la del algoritmo greedy.
- Llamemos G a la solución greedy y O la solución óptima.
- $G = g_1, g_2, ..., g_n$

Prueba

- Supongamos que el algoritmo no es óptimo.
- Entonces hay una solución mejor que la del algoritmo greedy.
- Llamemos G a la solución greedy y O la solución óptima.
- $G = g_1, g_2, ..., g_n$
- $O = o_1, o_2, ..., o_m$

Prueba

- Supongamos que el algoritmo no es óptimo.
- Entonces hay una solución mejor que la del algoritmo greedy.
- Llamemos G a la solución greedy y O la solución óptima.
- $G = g_1, g_2, ..., g_n$
- $O = o_1, o_2, ..., o_m$
- Para facilitar la notación diremos que las actividades de G y O están ordenadas por tiempo de finalización.

Prueba

- Si $g_1 \neq o_1$

Prueba

- Si $g_1 \neq o_1$
- Nota: tenemos que g_1 termina antes que o_1. (por como funciona el algoritmo)

Prueba

- Si $g_1 \neq o_1$
- Nota: tenemos que g_1 termina antes que o_1. (por como funciona el algoritmo)
- Entonces podemos formar una solución con esta pinta: $B = ((O - \{o_1\}) \cup \{g_1\})$ y esta solución es optima. (tiene tantas tareas como O)

Prueba

- Si $g_1 \neq o_1$
- Nota: tenemos que g_1 termina antes que o_1. (por como funciona el algoritmo)
- Entonces podemos formar una solución con esta pinta: $B = ((O - \{o_1\}) \cup \{g_1\})$ y esta solución es optima. (tiene tantas tareas como O)
- Ahora podemos probar lo mismo para el subproblema $G[2...n]$ $O[2...n]$

Por tanto el algoritmo es óptimo.

The Investor

The Investor

Un inversionista es exitoso si compra todas las empresas del mundo, las empresas cuestan una cierta cantidad de dinero y una vez adquiridas dan una cierta cantidad de dinero (ganancia fija). Diga si pepito es un inversionista exitoso.

Restricciones

Cantidad de empresas: $N < 10^5$ Dinero inicial: $N < 10^5$ Costo y ganancias de cada empresa: $c_i, g_i < 10^5$

Reflexiones

Reflexiones

Muchos algoritmos greedies sale con la idea del ordenamiento de las *tareas* y despues correr la heurística sobre ese orden.

- Algoritmo de Kruskal.
- El problema de la mochila con fraccionado.
- Assign Mice to Holes.
- Y muchos más...

Reflexiones

Reflexiones

Ahora bien, como ordenamos esto?. O mejor dicho que empresas compramos primero?

Reflexiones

Reflexiones

Ahora bien, como ordenamos esto?. O mejor dicho que empresas compramos primero?

- Sabemos que debemos comprar primero las empresas eficientes, estas son: las que den más ganancias que perdidas.

Reflexiones

Reflexiones

Ahora bien, como ordenamos esto?. O mejor dicho que empresas compramos primero?

- Sabemos que debemos comprar primero las empresas eficientes, estas son: las que den más ganancias que perdidas.
- Por otro lado debemos comprar las empresas que podamos afrontar. Que tengamos el dinero suficiente.

Solución

- Podemos ordenar con una función de comparación de la siguiente manera:
  - si la empresa es eficiente.
  - costo de la empresa.

Solución

- Podemos ordenar con una función de comparación de la siguiente manera:
  - si la empresa es eficiente.
  - costo de la empresa.
- Recorrer el arreglo sumando y restando el costo beneficio y si en algún momento no puedo comprar una empresa entonces no hay solución.

Solución

- Podemos ordenar con una función de comparación de la siguiente manera:
  - si la empresa es eficiente.
  - costo de la empresa.
- Recorrer el arreglo sumando y restando el costo beneficio y si en algún momento no puedo comprar una empresa entonces no hay solución.
- Pero, Qué hacemos con las empresas ineficientes?

Solución

- Pero, Qué hacemos con las empresas ineficientes?

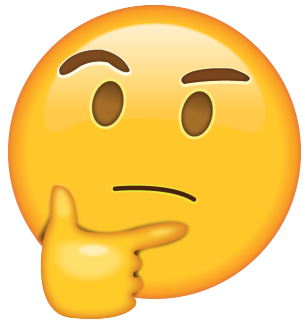
Solución

- Pero, Qué hacemos con las empresas ineficientes?
- Compramos las empresas más caras al principio?

Solución

- Pero, Qué hacemos con las empresas ineficientes?
- Compramos las empresas más caras al principio?
- Compramos las empresas más baratas al principio?

Solución



- Pero, Qué hacemos con las empresas ineficientes?
- Compramos las empresas más caras al principio?
- Compramos las empresas más baratas al principio?
- Compramos las empresas que den menos gap costo beneficio?

Solución

Compramos las empresas mas caras al principio

Contra-ejemplo

2 120
100 1
50 40

Solución

Compramos las empresas mas baratas al principio

Contra-ejemplo

2 120
100 70
50 1

Solución

Compramos las empresas que den menos gap costo-beneficio

Contra-ejemplo

2 120
120 70
50 49

Solución

Reflexiones

La mejor heurística es comprar las empresas que tiene mas g_i (ganancias). De esta manera usamos mejor las ganancias que nos dan estas empresas (pues la idea es comprar todas las empresas no la mayoría). Si compramos esas empresas estaremos en una mejor posición para comprar el resto.

Idea de la prueba

Idea de la prueba

La demostración es: tomemos dos empresas ineficientes, supongamos que las podemos comprar (digamos que haya solución). Luego probamos que en el orden creciente en las ganancias (nuestra idea) se ve que también funciona. Con cualquier solución podemos re-ordenar la compra de empresas ineficientes que estén en orden inverso de ganancias.

The Agency

En una galaxia lejana, cada planeta se representa como una secuencia de 0's y 1's. Viajar de un planeta a otro solamente es posible si difieren en una posición (de 101 podemos viajar al 100 pero desde el 001 no). Y el costo de visitar el planeta es el costo de aterrizaje. Que está dado por un costo fijo por cada elemento 1 de su secuencia. Por tanto el costo de ir a 0000 es 0 y el de ir a 1111 $\sum_{i=1}^{n} c_i$ Ahora bien. Cuánto es el mejor precio de ir desde A a B?

Ideas

- Ir al bit mas caro primero?

Ideas

- Ir al bit mas caro primero?
- Calcular todos los posibles secuencias y quedarme con la mejor?

Ideas

- Ir al bit mas caro primero?
- Calcular todos los posibles secuencias y quedarme con la mejor?
- Apagar los bit más caros que no me sirven primero y prender de los mas baratos a mas caros.

Ideas

- Ir al bit mas caro primero?
- Calcular todos los posibles secuencias y quedarme con la mejor?
- Apagar los bit más caros que no me sirven primero y prender de los mas baratos a mas caros.
- Prender los bit más baratos y luego apagar los mas caros.

Ideas

- Ir al bit mas caro primero?
- Calcular todos los posibles secuencias y quedarme con la mejor?
- Apagar los bit más caros que no me sirven primero y prender de los mas baratos a mas caros.
- Prender los bit más baratos y luego apagar los mas caros.
- Cualquier orden de prendido y apagado.

Solución

Ninguna de esas!

Reflexiones

Aveces vamos a tener que apagar algún bit que al final vamos a tener que prender, puesto que es tan costoso que conviene apagarlo y prenderlo de nuevo.

Formas de solucionarlo

Presentaremos dos formas de solucionarlo

- Algoritmo de Dijkstra. Es un algoritmo greedy para buscar caminos más cortos. Si se tiene una implementación eficiente el orden es $O(n * log(n))$

Formas de solucionarlo

Presentaremos dos formas de solucionarlo

- Algoritmo de Dijkstra. Es un algoritmo greedy para buscar caminos más cortos. Si se tiene una implementación eficiente el orden es $O(n * log(n))$
- Hacer una busqueda para apagar los i bits mas caros y luego aplicar la heurística: apagar los bit más caros que no me sirven primero y prender de los mas baratos a mas caros. Esta implementación es del $O(n^2)$

Forma de probar un algoritmo greedy

Forma de probar un algoritmo greedy

Escribirlo y correrlo contra los casos bases.

Forma de probar un algoritmo greedy

Na mentira!!. (Igual suele ser muy usado)

Forma de probar un algoritmo greedy

Forma de probar un algoritmo greedy

- Calcule la solución de su algoritmo y una solución general. Por ejemplo, sea $A = a_1, a_2, ..., a_k$ la solución de nuestro algoritmo y $O = o_1, ..., o_n$ el resultado óptimo.

Forma de probar un algoritmo greedy

Forma de probar un algoritmo greedy

- Calcule la solución de su algoritmo y una solución general. Por ejemplo, sea $A = a_1, a_2, ..., a_k$ la solución de nuestro algoritmo y $O = o_1, ..., o_n$ el resultado óptimo.
- Comparar greedy con otra solución. Suponga $A \neq O$ (ya que de lo contrario no hay nada que probar). Por lo general, puede aislar un ejemplo simple de esta diferencia, como uno de los siguientes:
  - Hay elementos de O que no están en A y elementos de A que no están en O.
  - Hay 2 elementos consecutivos en O en un orden diferente al de A (es decir, hay una inversión).

Forma de probar un algoritmo greedy

Forma de probar un algoritmo greedy

- Calcule la solución de su algoritmo y una solución general. Por ejemplo, sea $A = a_1, a_2, ..., a_k$ la solución de nuestro algoritmo y $O = o_1, ..., o_n$ el resultado óptimo.
- Comparar greedy con otra solución. Suponga $A \neq O$ (ya que de lo contrario no hay nada que probar). Por lo general, puede aislar un ejemplo simple de esta diferencia, como uno de los siguientes:
  - Hay elementos de O que no están en A y elementos de A que no están en O.
  - Hay 2 elementos consecutivos en O en un orden diferente al de A (es decir, hay una inversión).
- Intercambie los elementos en cuestión en O (ya sea intercambie un elemento de afuera y otro de adentro para el primer caso, o cambie el orden de los elementos en el segundo caso), y sostenga que tiene una solución que no es peor que antes.

Forma de probar un algoritmo greedy

Forma de probar un algoritmo greedy

- Hay que ver que si continuamos intercambiando, puedes eliminar todas las diferencias entre O y A en un número polinomial de pasos sin empeorar la calidad de la solución. Por lo tanto, la solución greedy producida es tan buena como cualquier solución óptima.

Forma de probar un algoritmo greedy

Forma de probar un algoritmo greedy

- Hay que ver que si continuamos intercambiando, puedes eliminar todas las diferencias entre O y A en un número polinomial de pasos sin empeorar la calidad de la solución. Por lo tanto, la solución greedy producida es tan buena como cualquier solución óptima.
- $G \rightarrow_p .... \rightarrow_p A$

Forma de probar un algoritmo greedy

Forma de probar un algoritmo greedy

- Hay que ver que si continuamos intercambiando, puedes eliminar todas las diferencias entre O y A en un número polinomial de pasos sin empeorar la calidad de la solución. Por lo tanto, la solución greedy producida es tan buena como cualquier solución óptima.
- $G \rightarrow_p .... \rightarrow_p A$
- Ojo con el uso de pruebas por el absurdo a partir de la suposicin $A \neq O$. El hecho de que la solución greedy no sea igual a la solución óptima seleccionada no significa que la codicia no sea óptima: podría haber muchas soluciones óptimas.

Un problema interesante

Con el aumento del uso de pesticidas, los arroyos y ríos locales se han contaminado tanto que se ha vuelto casi imposible para los animales acuáticos sobrevivir.
Frog Fred está en la orilla izquierda de un río y hay N rocas que están en línea recta desde la orilla izquierda a la orilla derecha. La distancia entre la orilla izquierda y la derecha es D metros. Hay rocas de dos tamaños. Las más grandes pueden soportar cualquier peso, pero los más pequeñas comienzan a undirse tan pronto como se coloca cualquier cosa. Fred tiene que ir a la orilla derecha, donde tiene que recoger un regalo y regresar a la orilla izquierda donde se encuentra su casa.
Puede aterrizar en cada roca pequeña una vez como máximo, pero puede usar las más grandes tantas veces como quiera. Nunca puede tocar el agua contaminada.
Se puede planificar los saltos de modo que se minimice la distancia máxima de un solo salto?

Restricciones

$T < 100$ número de test
$0 \leq N \leq 100$ cantidad de piedras.
$1 \leq D \leq 1000000000$ la distancia del río. Llamaremos S al conjunto de las piedras chicas y B al de las piedras grandes

Una forma de abordarlo

- Para abordarlo vamos a ver el problema para diferentes entradas.

Una forma de abordarlo

- Para abordarlo vamos a ver el problema para diferentes entradas.
- Si la cantidad de piedras $N = 0$ la respuesta es D.

Una forma de abordarlo

- Para abordarlo vamos a ver el problema para diferentes entradas.
- Si la cantidad de piedras $N = 0$ la respuesta es D.
- Probemos con la entrada siguiente: $S = [2, 3, 5, 7]$ $B = [4, 10]$ $D = 12$

Una forma de abordarlo

- Para abordarlo vamos a ver el problema para diferentes entradas.
- Si la cantidad de piedras $N = 0$ la respuesta es D.
- Probemos con la entrada siguiente: $S = [2, 3, 5, 7]$ $B = [4, 10]$ $D = 12$
- Probemos con la entrada siguiente: $S = [2, 3, 5, 7, 10]$ $B = [4, 6, 13]$ $D = 14$

Una forma de abordarlo

Una forma de abordarlo

- Una cosa que podemos observar es que no nos podemos gastar todas las piedras en el camino de ida (o de vuelta).

Una forma de abordarlo

- Una cosa que podemos observar es que no nos podemos gastar todas las piedras en el camino de ida (o de vuelta).
- Tampoco nos conviene consumir todas las piedras que están en una misma parte, ya que a la vuelta vamos a tener que dar un salto más grande.

Una forma de abordarlo

- Una cosa que podemos observar es que no nos podemos gastar todas las piedras en el camino de ida (o de vuelta).
- Tampoco nos conviene consumir todas las piedras que están en una misma parte, ya que a la vuelta vamos a tener que dar un salto más grande.
- Por tanto hay que tomar las piedras intercaladas.

Una forma de abordarlo

- Una cosa que podemos observar es que no nos podemos gastar todas las piedras en el camino de ida (o de vuelta).
- Tampoco nos conviene consumir todas las piedras que están en una misma parte, ya que a la vuelta vamos a tener que dar un salto más grande.
- Por tanto hay que tomar las piedras intercaladas.
- Link al problema: https://www.urionlinejudge.com.br/judge/en/problems/view/1054.

References

Preguntas?!