Universidad de Granada

Divide y Vencerás

Algorítmica

Lukas Häring García 2ºD

Tabla de contenidos

Ejercicio 1

Deseamos almacenar en una cinta magnética de longitud L un conjunto de n programas $P_1, P_2, \ldots, P_n$. Sabemos que cada P_i necesita un espacio a_i de la cinta y que $(\sum a_i) > L$, $1 \leq i \leq n$. Construir un algoritmo que seleccione aquel subconjunto de programas que hace que el número de programas almacenado en la cinta sea máximo.

1. **Conjunto de Candidatos**.
   Todos los programas.
2. **Función de Selección**.
   Seleccionamos aquel proceso que ocupe menos.
3. **Función Factible**.
   Si el espacio anteriormente ocupado $+a_k \leq L$.
4. **Objetivo**.
   Maximizar el número de procesos que entran en la cinta.
5. **Solución**.
   Se devuelve un vector $V = \{i_1, \cdots, i_k\}$ con todos los índices (sin importar el orden) de cada uno de los procesos.

En el peor de los caso, cuando el vector de procesos no está ordenado por tamaño, la eficiencia tiende a un orden $O(n^2)$ tras tener que comprar cada elemento con el anterior.
Formulación matemática.

$$\max \left\{ \sum_{j=1}^{k} P_{i_k} \right\} \leq L$$

Ejercicio 3

Se han de procesar n tareas con instantes de terminación t_i y tiempo de proceso p_i. Se dispone de un procesador y las tareas no son interrumpibles. Una solución factible planifica todas las tareas de modo que terminan en, o antes de, su instante asignado. Diseñar un algoritmo voraz para encontrar la solución.

1. **Conjunto de Candidatos**.
   Todas las tareas.

2. **Función de Selección**.
   La tarea que tenga menor tiempo de ejecución y sea factible.

3. **Función Factible**.
   Comprueba que si en el instante actual a y el tiempo que tarda en ejecutarse el proceso p_i es menor al tiempo de terminación $a + p_i < t_i$.

4. **Objetivo**.
   Ejecutar el máximo de tareas antes del instante asignado.

5. **Solución**.
   Un vector $V = \{i_1, \cdots, i_k\}$ que contiene el índice de la tarea a asignar (Cola).

La eficiencia del algoritmo es $O(n^2)$ ya que debe buscar todo aquel tiempo de proceso mínimo y que el instante actual sea inferior al de terminación más el de ejecución.
Formulación matemática.

Ejercicio 4

Dado un conjunto de n cintas cuyo contenido está ordenado y con ni registros cada una, se han de mezclar por pares hasta lograr una única cinta ordenada. La secuencia en que se haga la mezcla determina la eficiencia del proceso. Diseñar un algoritmo voraz que minimice el número de movimientos.

1. **Conjunto de Candidatos**.
   Todas las cintas.

2. **Función Factible**.
   No tiene.

3. **Función de Selección**.
   Se selecciona aquella cinta no utilizada anteriormente y menor a todas las demás que tampoco han sido utilizadas.

4. **Objetivo**.
   Minimizar el número de movimientos entre las cintas.

5. **Solución**.
   Una cola ordenada según que elemento se coge y el siguiente es aquel que se vierte.

Como hay insertar todo aquel elemento maximal a los insertados anteriormente, su eficiencia es $O(n^2)$.

Ejercicio 5

Un automovilista ha de ir con su vehículo desde la población A hasta la población B siguiendo una ruta prefijada (por ejemplo la carretera N-340). Con el depósito completamente lleno puede hacer X km. El conductor conoce de antemano en qué lugares de la carretera existen gasolineras (la distancia entre dos gasolineras consecutivas es inferior a X km.). Diseñar un algoritmo que permita que el automovilista vaya de A a B repostando el mínimo número de veces posible (se supone que parte de A con el depósito lleno y que el coche no se averiará, ni tendrá un accidente, ni será abducido, etc. durante el trayecto).Indicar el coste del algoritmo

1. **Candidatos**.
   Todas aquellas gasolineras que están en el camino del automovilista.

2. **Función de Selección**.
   Seleccionamos aquella gasolinera que está más lejos de las demás alcanzables por el coche y en la dirección correcta.

3. **Función Factible**.
   Verifica si la gasolinera es alcanzable con la gasolina restante.

4. **Objetivo**.
   Pasar por el mínimo número de gasolineras.

5. **Solución**.
   Un vector de gasolineras por donde el conductor debe pasar.

Si existen n gasolineras y cada una está a la distancia $(X - 1)$km de la anterior, por lo que debemos pasar por todas las ellas, es decir, $O(n)$.

Ejercicio 7

Dado un grafo $G =< N, A >$ siendo N el conjunto de vértices y A el conjunto de aristas. Se pide colorear el grafo de forma que si dos vértices están unidos por una arista los colores asignados a estos vértices tienen que ser distintos. El objetivo en este problema es usar el menor numero de colores para colorear todos los vértices del grafo.

1. **Conjunto de Candidatos**.
   Todos los vértices del grafo.
2. **Función de Selección**.
   Se seleccionan un vértice no coloreado, cualquiera.
3. **Función Factible**.
   Verifica si el color asignado no coincide con los que tienen sus vecinos.
4. **Objetivo**.
   Usar el menor número de colores.
5. **Solución**.
   Un vector de colores para cada vértice, tal que el vértice v_i tiene un color c_i.

Cuando el grafo (de n vértices) tiene el mismo número de vértices que de vecinos (esto ocurre cuando es un grafo completo K_n), se debe comprobar el color con el de los vecinos. La eficiencia tiende a $O(n^2)$.

Especificaciones

1. Windows 10.0.14393
2. Procesador Intel(R) Core(TM) i7-7800X CPU @ 3.50GHz, 3504 Mhz
3. 6 procesadores principales.
4. 12 procesadores lógicos.
5. Memoria física instalada (RAM) 8,00 GB x 2
6. Compilador MinGW.