Algoritmia básica

Índice

Capítulo 1

ALGORITMOS VORACES

Los algoritmos voraces se utilizan para resolver problemas de optimización: de un **conjunto de elementos candidatos**, se quiere obtener un **subconjunto solución factible** que maximice/minimice una **función objetivo**.

1. Se inicia con un conjunto vacío de candidatos.
2. Se intenta añadir el mejor elemento no escogido siguiendo una **función de selección**.
3. Si añadiendo el elemento el problema aún es **completable** (se puede llegar a la solución añadiendo mas elementos) se añade, de lo contrario se rechaza y elimina como posible elemento.
4. Si no es solución, volver al paso 2

1.1 El problema de la mochila

Se tiene una mochila con capacidad limitada (C), y una serie de objetos fraccionables (n) con un peso determinado(p_i). Meter una fracción de un objeto (x_i) da un beneficio ($b_i x_i$). Se quiere llenar la mochila con el máximo beneficio sin superar el peso ($(x_1, ...x_n)$ tal que $max(\sum_{1 \leq i \leq n} b_i x_i)$ y $\sum_{1 \leq i \leq n} p_i x_i \leq C$).

La estrategia mas óptima es tomar los objetos que proporcionen mayor beneficio por unidad de peso.

```
const int n = /* Número objetos */

/*
    Pre: Los objetos deben estar ordenados de mayor a menor según la proporción
         de beneficio obtenido por unidad de peso:
         ∀i∈1..n:peso[i]>0 ∧ ∀i∈1..n-1:benef[i]/peso[i]≥benef[i+1]/peso[i+1]
    Post: sol es solución óptima del problema de la mochila
 */
double[n] mochila(double[n] benef, peso, double cap)
{
    double[n] sol = {}; double resto = cap; int i = 1;

    for (i; i ⩽ n && peso[i]⩽ resto; i++)
    {
        sol[i] = 1;
        resto -= peso[i];
    }

    if (i ⩽ n) sol[i] = resto/peso[i]

    return sol;
}
```

1.2 Caminos mínimos en grafos (Dijkstra)

1.3 Árboles de recubrimiento de coste mínimo

1.4 Códigos de Huffman

1.5 El problema de la selección de actividades

1.6 El problema de la minimización del tiempo de espera

1.7 Fundamentos teóricos del esquema voraz

1.8 Un problema de planificaciónn de tareas a plazo fijo

1.9 Heurísticas voraces

1.9.1 Coloreado de grados

1.9.2 El problema del viajante de comercio

Capítulo 2

DIVIDE Y VENCERÁS

Capítulo 3

PROGRAMACIÓN DINÁMICA

Capítulo 4

BÚSQUEDA CON RETROCESO

Capítulo 5

RAMIFICACIÓN Y PODA

Capítulo 6

PROGRAMACIÓN LINEAL Y REDUCCIONES