

Proyecto Algoritmos

Camilo Gómez V. Santiago Ortiz.

Universidad del Rosario

October 2019

Introducción

Dado un plan de estudios realizado por una coordinación académica para un estudiante, al momento de perder, cancelar o adelantar una materia, el plan de estudios pierde su orden y su replanteamiento no es sencillo.

Problema

De acuerdo a la malla curricular de una carrera universitaria, se quiere optimizar la carga académica semestral de acuerdo a ciertos parámetros de cada materia, que son:

- ▶ numero de credits
- ▶ requisitos
- ▶ correquisitos
- ▶ prioridad
- ▶ etiqueta

Problema

El objetivo principal es establecer un pensum, que dependiendo de la cantidad de créditos matriculable, se plante la mejor distribución de las materias, desarrollando la carrera en el menor tiempo posible, la mejor opción para un estudiante ideal.

Implementación

Introducción - Algoritmos voraces

Son algoritmos de implementación simple utilizados en su mayoría para resolver problemas de optimización, por ejemplo, encontrar la distribución y secuencia óptima de las materias dentro de una malla curricular

- ▶ entrada: conjunto finito de n elementos
- ▶ salida: conjunto S que cumple con los parámetros y restricciones tal que minimiza o maximiza el problema inicial, dicho conjunto se le llama solución óptima.

elementos de un algoritmo voraz

- ▶ conjunto de elementos con los posibles candidatos
- ▶ función que determina si el conjunto de candidaos es una solución al problema
- ▶ función que determina si el conjunto aún se puede completar
- ▶ función de seleccion
- ▶ función objetvio que da el valor de una solución según lo que se pretende optimizar.

Estructuras de datos

Subject

Parámetros:

- ▶ nombre
- ▶ credits
- ▶ prerequisito
- ▶ equiqueta

Métodos:

- ▶ get_priority
- ▶ get_name
- ▶ get_request
- ▶ get_num_credits
- ▶ get_label

Operadores:

- ▶ <
- ▶ <<
- ▶ ==
- ▶ ()

Estructuras de datos

Curriculum

Subestructura:

- ▶ tot_credits
- ▶ vector
- ▶ punteros:
next_semester,
previous_semester
- ▶ función auxf
- ▶ operador =

Metodos:

- ▶ insert
- ▶ search
- ▶ display
- ▶ maker

métodos privados:

- ▶ create_new_semester
- ▶ set_insert_control
- ▶ aux_insert
- ▶ aux_insert2

Algoritmos implementados

maker

Implementamos un algoritmo voraz de nombre maker, que consta de distintas partes:

- ▶ priority queues
- ▶ vector
- ▶ insert