Tehnica Greedy

Tehnica *Greedy* este una dintre cele mai directe tehnici de proiectare, aplicându-se mai mult la probleme de optimizare. Aceasta presupune că probleme ce trebuie rezolvate au structura următoare:

* o mulţime A={a1, a2,...,an}, din n elemente
* se cere determinarea unei submulţimi B, care aparţine mulţimii A, ce îndeplinește anumite condiţii pentru a putea fi acceptată o soluţie.

*Greedy* constă în faptul că soluţia se gasește pas cu pas, de fiecare dată fiind selectată în soluţie elementul care pare cel mai bun, sperând sa aducă la o soluţie globală.

În cazul dat, submulţimea B și, la rândul lor, submulţimile lui B poate fi considerată ***soluţie posibilă,*** dacă urmăresc un criteriu de selecţie formulat de prorgramator. Din mulţimea tuturor soluţiilor posibile, se va alege o ***soluţie optimă***.

Numele “Greedy” vine de la faptul că soluţiile posibile sunt “înghiţite”, acestea fiind elementele “gustoase” a mulţimii A, creând o imagine al unui program “lacom”.

Un lucru important de reţinut ar fi că Metoda *Greedy* caută *sa introducă direct un element x în soluţia optimă* și nu să determine toate soluţiile și s-o aleagă pe cea mai optimă din acestea, lucru ce sporește eficienţa algoritmilor *Greedy* în cazul unui număr prea mare de soluţii posibile.

**Schema generală:**

**while** ExistaElemente **do**

**begin**

AlegeUnElement(x);

IncludeElementul(x);

**end**

**Avantajele** Metodei *Greedy* ar fi că poate fi aplicată mai multor probleme, precum:

* codificarea arborilor Huffmann
* determinarea arborelui minimal de acoperire
* planificarea activităţilor
* problema fracţionară a rucsacului
* problema spectacolelor

Totodată, e mai rapidă decât *Backtracking*.

**Dezavantajele** ar fi că aceasta tehnică are nevoie de un anumit algoritm existent pentru a putea fi implimentată. Pe lângă asta, singura soluţie care e aleasă de acesta, nu e întodeauna globală.