

Dokumentacia - interval-tree.cpp

Objektovo orientovana implementacia suctoveho intervaloveho stromu.

class **interval_tree**

obsahuje:

- private
 - class vertex
 - pointer na object vertex (koren stromu)
- konstruktor - berie 1 argument s (velkost postupnosti ktora sa bude ukladat do stromu), ktory priradi premennej size. Najde najblizsiu vacsiu mocninu 2 nez je s (ulozi do n) a nastavi root na new Vertex(0, n)
- destruktor - zmaze root
- public int size (velkost postupnosti ktora sa uklada do listou stromu) a funkcie:
 - change(int i, int v)
 - sum(int l, int r)
 - change_interval(int l, int r, int v)

Kazda z tychto funkcii zavola do korena prislusnu funkciu z triedy vertex

class **vertex**

- private:
 - int value - hodnota v danom vrchole
 - int lazy - hodnota lazy, ktoru vrchol pokym nemusí tak neposuvá ďalej
 - int begin, end - začiatok a koniec intervalu o ktorý sa stara daný vrchol
 - vertex *left, *right - ukazovatele na synov
- konstruktor - nastavi všetky potrebné premenné pre daný vrchol a zavola sa rekurzívne do synov. Celý strom má po vytvorení vo vrchole value = 0 a lazy = 1
- funkcie:
 - send_info() - ak má vrchol hodnotu lazy roznu od 1 tak ju pošle synom a nastavi si ju na 1. Aj pravému aj ľavému synovi sa k ich hodnotám lazy prinasobí lazy otca a ich value sa touto hodnotou prenasobí aby mali správnu hodnotu uloženú vo svojom vrchole.
 - change(int i, int v) - rekurzívne sa vnára až do listu ktorý je zodpovedný za i-ty prvok postupnosti pričom vždy volá funkciu send_info(). Ak príde do daného listu zmení mu value=v
 - sum(int l, int r) - listy pod daným vrcholom:
 - ležia všetky na intervale [l,r) - vráti sa value vrchola
 - neležia vôbec na [l,r) - vráti sa 0
 - ležia z časti - zavola sa funkcia send_info() a vráti sa hodnota sučtu funkcie sum, ktorú vrátia synovia daného vrcholu
 - change_interval(int l, int r, int v) - ak daný vrchol:
 - sa stara o interval [l,r) - prenasobí sa mu lazy a value hodnotou v
 - nestara sa vôbec o [l,r) - nerobí sa nič
 - stara sa z časti o [l,r) - zavola sa rekurzívne do synov

Kazda z funkcií change, sum a change_interval beží asymptoticky v $O(\log n)$, kde n je najbližšia väčšia mocnina 2 než je veľkosť vstupnej postupnosti.