

README.md

Alex Falzone

January 2021

1 Introduction

Creata nel 1975 dall'informatico Peter van Emde Boas.

Questo albero ha la particolare caratteristica di svolgere le operazioni di search, insert, delete, minimum, maximum, successor, predecessor nel tempo $O(\lg \lg u)$ dove u rappresenta la dimensione dell'universo.

Possiamo pensare a quest'albero come un array A (diviso in \sqrt{u} parti) contenenti 0 o 1. Definito un array **summary** $[0 \dots \sqrt{u} - 1]$, dove **summary** $[i]$ contiene 1 se e soltanto se il sottoarray $A[i\sqrt{u} \dots (i+1)\sqrt{u} - 1]$ contiene almeno un 1. Questo sottoarray di A di \sqrt{u} bit è detto **i-esimo cluster**.

2 Struttura dell'albero

Indichiamo con $\sqrt[4]{u}$ la radice quadrata **inferiore**, ovvero

$$2^{\lfloor \lg(u)/2 \rfloor}. \quad (1)$$

Mentre con $\sqrt[4]{u}$ la radice **superiore**, ovvero

$$2^{\lceil \lg(u)/2 \rceil} \quad (2)$$

Denotiamo con $\text{vEB}(u)$ un albero **vEB** con dimensione dell'universo pari a u e, a meno che u non sia uguale alla dimensione base 2, l'attributo **summary** punta a un albero $\text{vEB}(\sqrt[4]{u})$ e l'array **cluster** $[0 \dots \sqrt[4]{u} - 1]$ punta ai $\sqrt[4]{u}$ alberi $\text{vEB}(\sqrt[4]{u})$.

Inoltre all'interno di un cluster è presente un valore **min** e un valore **max**. Banalmente memorizzano rispettivamente l'elemento minimo e massimo nell'albero **vEB**. Inoltre è importante notare che l'elemento memorizzato in **min** non appare in nessuno degli alberi di ricorsione $\text{vEB}(\sqrt[4]{u})$ cui punta l'array **cluster** (diversamente accade per **max**).

Successivamente è necessario stabilire come accedere sia al numero di cluster di un determinato valore(x), sia alla posizione del valore(x) all'interno del cluster; Esso avviene usando rispettivamente:

$$\text{high}(x) = \lfloor x / \sqrt[4]{u} \rfloor \quad (3)$$

$$low(x) = x \bmod \sqrt[u]{u} \quad (4)$$

$$index(x, y) = (x \sqrt[u]{u}) + y \quad (5)$$

Dove index rappresenta il numero(x) e la posizione(y) del cluster.

2.1 Operazioni vEB

2.1.1 Minimo e massimo

Algorithm 1 Minimum(V)

return V.min

Algorithm 2 Maximum(V)

return V.max

Poiché il massimo e il minimo sono memorizzati negli attributi **min** e **max** essi richiedono tempo costante, ovvero $O(1)$.

2.1.2 Successore e Predecessore