Lorenzo Bellomo 531423

- avg height di un Treap di n chiavi e priorità random è $O(\log n)$

definisco ~~X~~ $A_k^i = \begin{cases} 1 & \text{se } i \text{ è ancestor di } k \\ 0 & \text{altrimenti} \end{cases}$

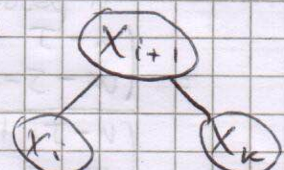
$depth(x_k) = \sum_{i=1}^{n} A_k^i$

e partire da adesso - $p(x)$ è probabilità che evento x accade
- $pr(x_i)$ è la priorità della chiave x_i

$\{x_i, x_{i+1} \ldots x_k\}$ sia dato un insieme di questo tipo, in cui le chiavi $x_i < x_{i+1} \ldots < x_k$, quindi crescente

se $pr(x_{i+1}) < pr(x_i)$ e $pr(x_{i+1}) < pr(x_k)$

allora verrà generato un treap di questo tipo



$\Rightarrow$ Per far si che $A_k^i = 1$, allora x_i deve essere l'elemento a priorità minima nel set definito (dimensione $|k-i|+1$)

$E[depth(x_k)] = \sum_{i=1}^{n} E[A_k^i] = \sum_{i=1}^{n} p(A_k^i = 1)$ ← utilizzo adesso ip. che priorità siano random

$\Rightarrow \sum_{i=1}^{n} \frac{1}{|k-i|+1} = O(\log n)$

- r nodes, n edges Cuckoo graph $p(\text{(i)} \xrightarrow{L} \text{(J)}) \le \frac{c^{-L}}{r}$

con $r \ge 2cn$, quindi $n \le \frac{r}{2c}$

Prova per induzione su L

caso base: $L = 1$

① $h_1(i) = h_2(J)$ or $h_2(i) = h_1(J)$

$P(\text{(i)} \to \text{(J)}) \overset{①}{=} \frac{2}{r^2} \cdot n \le \frac{2}{r^2} \cdot \frac{r}{2c} = \frac{1}{cr} = \frac{c^{-1}}{r}$

caso induttivo: L

Ip induttiva $p(\text{(i)} \xrightarrow{L-1} \text{(h)}) \le \frac{c^{-(L-1)}}{r}$

l'ho già calcolato, è $p(\text{(h)} \to \text{(J)})$

$P(\text{(i)} \xrightarrow{L-1} \text{(h)} \to \text{(J)}) \le \sum_h \frac{c^{-(L-1)}}{r} \cdot \frac{c^{-1}}{r} = r \cdot \frac{c^{-L}}{r^2} = \frac{c^{-L}}{r}$

Streaming Sampling

```
s = 0;   // # of selected items
for j = 1 to n {
    h = rand(0,1);
    if (h ≤ (m−s)/(n−j+1)) {
        output S[j];
        s++;
    }
}
```

È chiaro che la correttezza dipende dal valore della guardia dell'if.

s è il numero di elementi già selezionati.

Se $s = m$, quindi ho già scelto m elem allora l'algoritmo non seleziona più elementi, viceversa se ne rimangono k e ne devo selezionare k allora li seleziona tutti.

sia $x = \frac{m-s}{n-j+1}$ ✓ il valore con cui confronto h

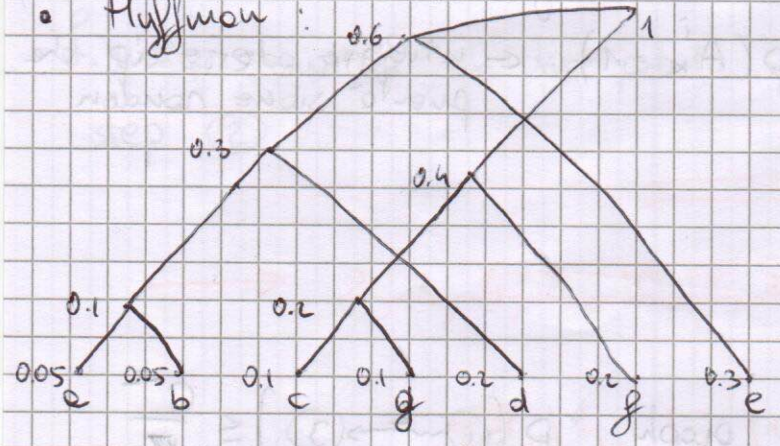
allora

$$\frac{\#\text{ possibilità di scegliere } m-s \text{ elementi compreso } S[j]}{\#\text{ possibilità di scegliere } m-s \text{ elementi}} = \cancel{\#(\text{sample } S[j]) \text{ over } m\text{-}s}$$

$$= \frac{\binom{n-j}{m-s-1}}{\binom{n-j+1}{m-s}} = \frac{\frac{(n-j)!}{(m-s-1)!\,(n-j-m+s+1)!}}{\frac{(n-j+1)!}{(m-s)!\,(n-j+1-m+s)!}} = \frac{m-s}{n-j+1}$$

⇐ esattamente la guardia dell'if

- Huffman:



Costruisco symbol table, ncw e fcw array

symb / level	1	2	3
1			
2	f	e	
3	c	g	d
4	a	b	

	1	2	3	4
ncw	0	2	3	2
fcw	2	2	1	0

Droppo ncw e l'albero

⇒ uso soltanto fcw e symb

T = 1000 1011 0 11100. _

primo simbolo:

parto da lvl 1 ⇒ $(1)_2 < 2 = FCW[1]$

⇒ $(10)_2 \geq 2 = FCW[2]$

prendo symb[lvl = 2, cwd = 1] = f

primo simbolo = f

continuo il decoding

$(0)_2 < 2 = FCW[1]$

$\Rightarrow (00)_2 < 2 = FCW[2]$

$\Rightarrow (001)_2 \geq 1 = FCW[3]$

symb [lvl = 3, cwd = 1] = c

secondo simbolo = c

continuo il decoding

$(0)_2 < 2 = FCW[1]$

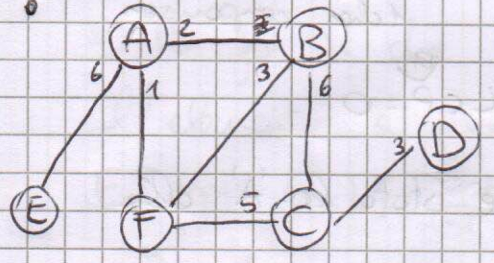
$\Rightarrow (01)_2 < 2 = FCW[2]$

$\Rightarrow (011)_2 \geq 1 = FCW[3]$

symb [lvl = 3, cwd = 3] = d

$T = \underbrace{10}_{f}\ \underbrace{001}_{c}\ \underbrace{011}_{d}$

•



I sort the edges according to the weigth and to the first component

Sorted = {(A, F, 1), (A, B, 2), (B, F, 3), (C, D, 3), (C, F, 5), (A, E, 6), (B, C, 6)}

estraggo gli edges in ordine

$\Rightarrow$ (A, F, 1) $\longrightarrow$ MST

$\Rightarrow$ (A, B, 2) $\longrightarrow$ MST

$\Rightarrow$ (B, F, 3) NOT to MST, si crea un ciclo

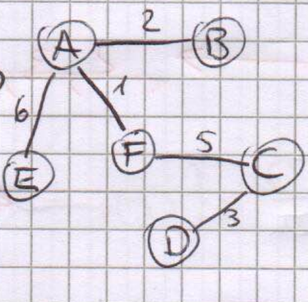
$\Rightarrow$ (C, D, 3) $\longrightarrow$ MST

$\Rightarrow$ (C, F, 5) $\longrightarrow$ MST

$\Rightarrow$ (A, E, 6) $\longrightarrow$ MST

$\Rightarrow$ (B, C, 6) NOT to MST, already n-1 edges selected

MST $\Rightarrow$



• $S = \underset{1\,2\,3\,4\,5\,6\,7}{BAOBAB\$}$ appendo dollaro, i numeri cerchiati indicheranno l'ordine di computazione di tale LCP

L'array SUFF è solo per comodità di visualizzazione

	LCP	SA	SUFF
5	0	7	$
2	1	5	AB$
6	0	2	AOBAB$
4	1	6	B$
1	2	4	BAB$
3	0	1	BAOBAB$
	/	3	OBAB$

1 : confronto BAOBAB$
BAB$
3 char comparison => LCP = 2

2 : confronto AB$
AOBAB$

A non deve essere confrontato, in quanto LCP ≥ 2-1=1
=> 1 char comparison ('B' e 'O') => LCP = 1

3 : confronto OBAB$
BAOBAB$

LCP = 0 ⇐ 1 char comparison

4 : confronto BAB$
B$

2 char comparison => LCP = 1

5 : confronto AB$
$

1 char comparison => LCP = 0

6 confronto B$
AOBAB$

1 char comparison

LCP = 0

L'unico carattere che ho potuto non confrontare è stato la 'A' allo step (2)