ESERCIZIO 4

S = baobab

Suffix Array

baobab$
aobab$
obab$
bab$
ab$
b$
$

→

	SA	SApos	LCP	CMP
$	7	1	0	1
ab$	5	2	1	1
aobab$	2	3	0	1
b$	6	4	1	2
bab$	4	5	2	3
baobab$	1	6	0	1
obab$	3	7	/	/

B A O B A B $
1 2 3 4 5 6 7

Algoritmo LCP in $O(n)$

① $i = 1$ $h = 0$ $q = 6$ $k = 4$

$T[4+0, 1+0]$

② $i = 2$ $h = 1$ $q = 3$ $k = 5$

$T[5+1, 2+1]$

③ $i = 3 \quad h = 0 \quad q = 7 \quad k = 1$

$T[1+0, 3+0]$

④ $i = 4 \quad h = 0 \quad q = 5 \quad k = 6$

$T[6+0, 4+0]$

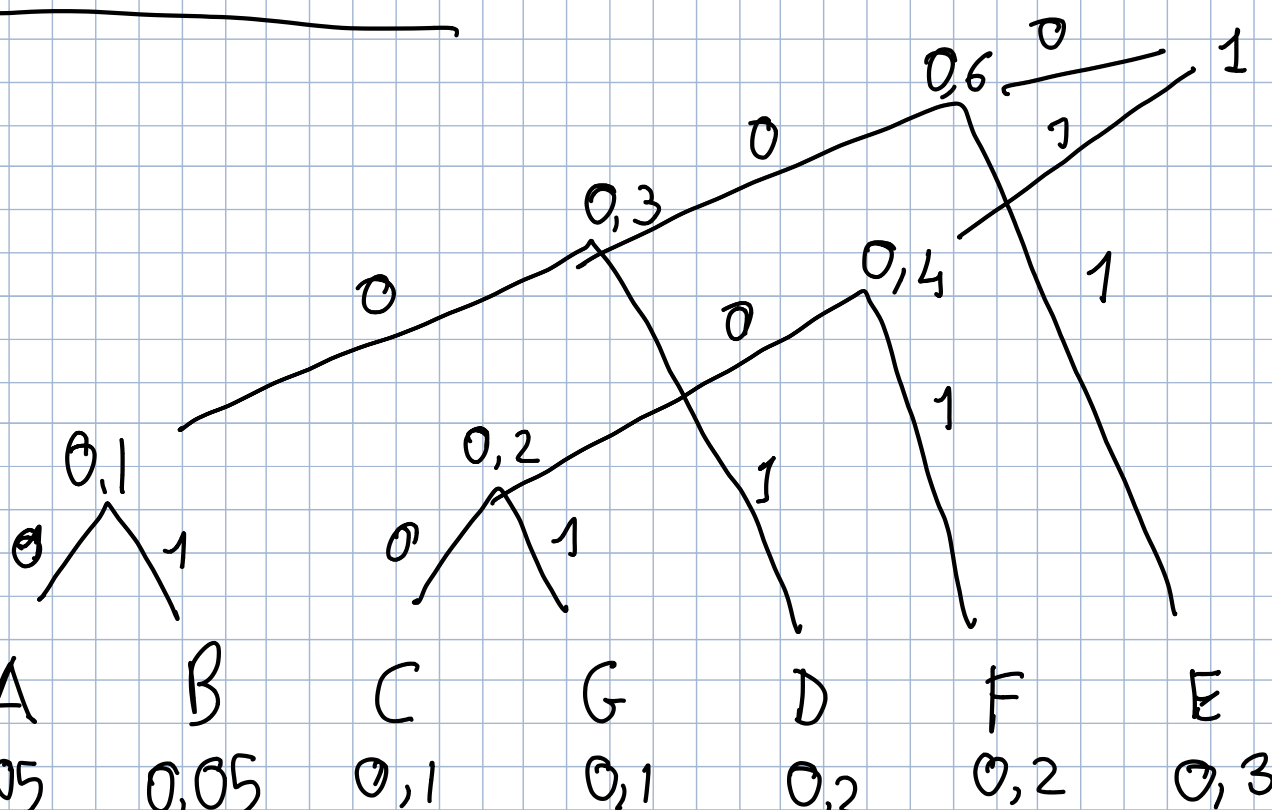
⑤ $i = 5 \quad h = 0 \quad q = 2 \quad k = 7$

$T[7+0, 5+0]$

⑥ $i = 6 \quad h = 0 \quad q = 4 \quad k = 2$

$T[2+0, 6+0]$

ESERCIZIO 2



num

1	2	3	4
0	2	3	2

fc :

1	2	3	4
2	2	1	0

$2 \to 10$, $1 \to 001$, $0 \to 0000$

symb

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

Decode $\underbrace{10}\,\underbrace{0\,0\,1}\,\underbrace{0\ 1\ 1}\,0\,1\,1\,1\,0\,0$

① · $V = 1 \quad l = 2$

$V < fc[1]$ si

$V = 2 + 0 = 2$

· $V = 2 \quad l = 2$

$V < fc[2]$ NO

$Symb[\overset{l}{2}, \overset{l}{2} - \overset{fc[l]}{2}] = [2, 0]$

$10 = f$

② · $V = 0 \qquad l = 1$

$V < fc[1]$ sì

- $V = 0 \quad l = 2$

  $V < fc[2]$ sì

- $V = 1 \quad l = 3$

  $V < fc[3]$ NO

$Symb[l, V - fc[l]] =$

$[3, 0]$

$001 = c$

3
- $V = 0 \quad l = 1$

  $V < fc[1]$ sì

- $V = 1 \quad l = 2$

  $V < fc[2]$ sì

- $V = 3 \quad l = 3$

$v < f_{c[2]}$ NO

$$\text{Symb}[3, 3-1] = [3,2]$$

$$O11 = d$$

ESERCIZIO 1

(2) Prob that i,j connected by a path of length L is at most $\frac{c^{-L}}{r}$ dove $r \geq 2cn$

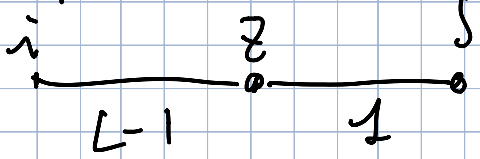
Dim. prendiamo il caso $L = 1$
In questo caso abbiamo che una chiave finisce in i o j con probabilità $\frac{1}{m}$. o perché ce la manda h_1 o perché ce la manda h_2.
Quindi abbiamo $\frac{2}{r^2}$ probabilità, se consideriamo poi n chiavi allora abbiamo

$$\sum_{k \in S} \frac{2}{r^2} = \frac{2n}{r^2}$$ ma dato che $r \geq 2cn$

e $2n < r$

$$\frac{r}{Cr^2} = \frac{C^{-1}}{r}$$

Per il caso $L \geq 1$ abbiamo un percorso lungo L lo spezziamo in due (per $L-1$ il teorema vale)



Ora diciamo che nel caso

→ $L-1$ abbiamo una probabilità $\frac{C^{-L+1}}{r}$

→ Nel caso $L=1$ abbiamo $\frac{C^{-1}}{r}$

Li moltiplichiamo: $\frac{C^{-L}}{r^2}$

Consideriamo r valori possibili per z e otteniamo

$\frac{C^{-L}}{r}$ che è la prob. che due nodi siano connessi da un arco di lunghezza L.

(1) Prove that the average height of a treap on n keys with random priorities is $O(\lg n)$

Dim: Consideriamo le chiavi da inserire nel treap $K = \{x_1 \dots x_k\}$ con $x_i < x_{i+1}$, $x_{i+1} < x_k$

e ognuna con una priorità presa random.

Definiamo

$$A_k^i = \begin{cases} 1, & \text{se } x_i \text{ è ancestor di } x_k \\ 0 & \text{altrimenti} \end{cases}$$

La profondità a cui si trova un certo nodo i dipende da quanti ancestor ha

$$\text{depth}(k) = \sum_{i=1}^{n} A_k^i = \sum_{i=1}^{n} P(x_i \text{ ancestor di } x_k) \atop X_i = 1$$

Con ancestor intendiamo che preso il range di chiavi $X(1,k)$ allora i è il nodo che ha la priorità minore del range.

Quindi

$$E[\text{depth}(k)] = E\left[\sum_{i=1}^{n} P(X_i = 1)\right]$$

Dobbiamo calcolare la probabilità che $X_i = 1$ ovvero che x_i sia il minimo del range.

Abbiamo $\frac{1}{}$ prob. che sia il minimo

$|k-i|+1$

quindi vuol dire

$$E[\text{depth}(k)] = E\left[\sum_{i=1}^{n} \frac{1}{|k-i|+1}\right] = O(\log n)$$

③ Streaming algorithm to sample u.a.r m item da uno stream lungo n. Provare la correttezza.

$S = 0$

for j in $S[1, n]$ && $(s < m)$:

$\quad p = \text{Rand}(0,1)$

$\quad$ if $\left(p < \frac{m-s}{n-j+1}\right)$:

$\quad\quad$ Sample j

$\quad\quad$ S++

m = punto in cui sono arrivato

S = quanti sample ho

Provare che la prob è $\frac{1}{n}$.

Nel caso in cui $m = 1$ abbiamo

$P(1) = \frac{1}{N}$, $P(2) = \frac{1}{N-1}$, ... $P(N) = \frac{1}{\cancel{N}} = 1$

Per un generico J abbiamo

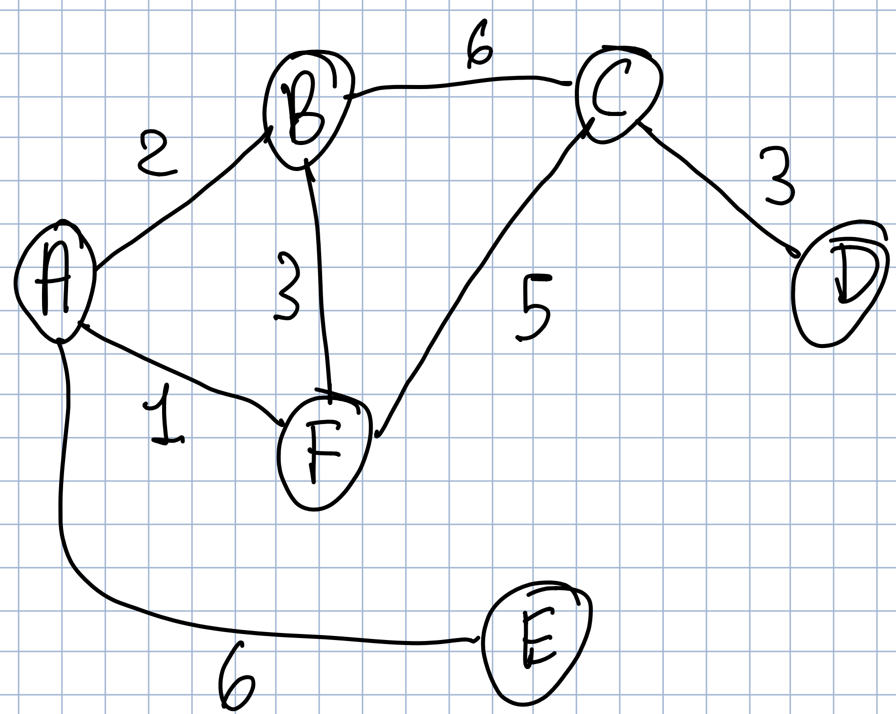
$P(J) = \frac{1}{N-J+1}$ ora se siamo allo step

$N-J+1$ N e per i precedenti
$N-1$ vale il teorema e
ho prob $\frac{1}{N}$ uniforme di prendere un intero come sample allora abbiamo che la prob di non prendere i precedenti $N-1$ è $\left(1-\frac{J-1}{N}\right)$

$$P(\text{sample } J) = P(\text{not sample } J-1) \cdot P(\text{sample } J)$$

$$\left(1-\frac{J-1}{N}\right) \cdot \left(\frac{1}{N-J+1}\right) = \frac{1}{N}$$

ESERCIZIO 3



MST con algoritmo di Kruskal:
Dobbiamo ordinare gli archi per ordine crescente e poi ogni volta prendiamo il minimo, controlliamo che prendendolo non si crei un ciclo e allora in quel caso consideriamo l'arco:

(A, F, 1)
(A, B, 2)
(B, F, 3) NO
(C, D, 3)
(C, F, 5)
(A, E, 6)
(B, C, 6) NO