L16: Backwards analysis

10/19/2021

$\{1, 2, \ldots, n\} = [n]$

π be a random permutation of $[n]$

$\pi_i \in [n]$ $\quad \pi_1, \pi_2, \ldots$

$X_i = 1 \iff \pi_i < \pi_1, \pi_2, \ldots \pi_{i-1}$

$\pi_i = \min(\pi_1, \ldots, \pi_i)$

7, 1, 3, 5, 4, 2, 6

2

$X_1 = X_2 = 1$

$Y = \sum_{i=1}^{n} X_i$ $\quad E[Y] = E[\sum X_i] = \sum E[X_i]$

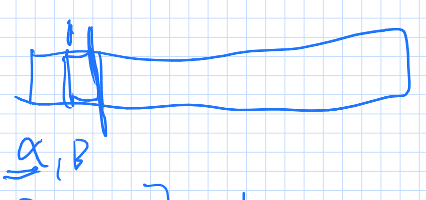
$= \sum_{i=1}^{n} P[X_i = 1] = \sum_{i=1}^{n} \frac{1}{i} \leq \ln n + \gamma$

$\ln n + O(1)$

$O(\log n)$

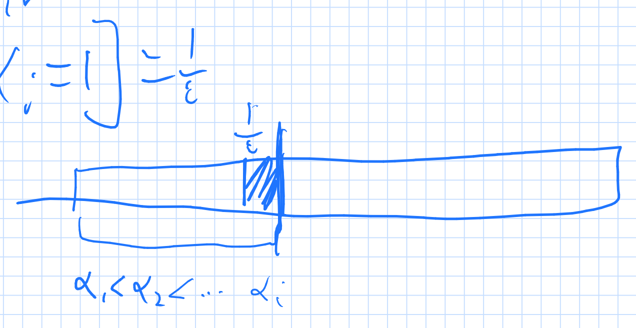
$P[X_1 = 1] = 1$

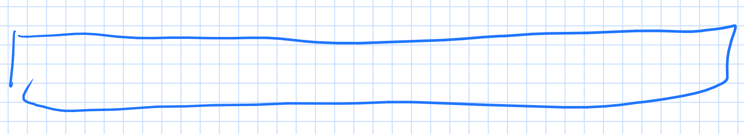
$P[X_2 = 1] = \frac{1}{2}$



α, β

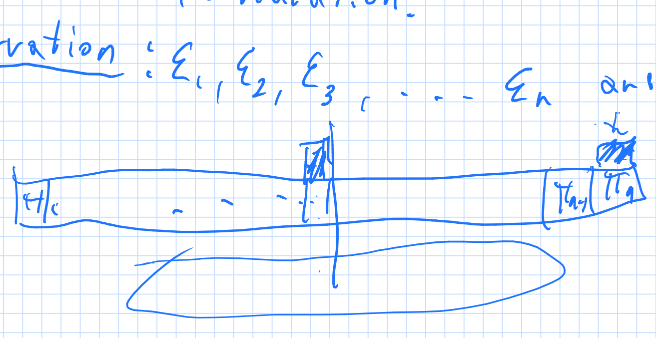
$P[X_i = 1] = \frac{1}{i}$





$\varepsilon_i = \pi_i$ is the min of first i element in the random permutation.

Observation: $\varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots \varepsilon_n$ are independent.



---

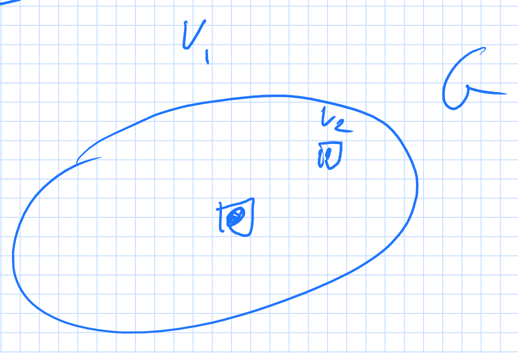
prob[# min change $> t \ln n$] "$\leq$" $\frac{1}{n^t}$

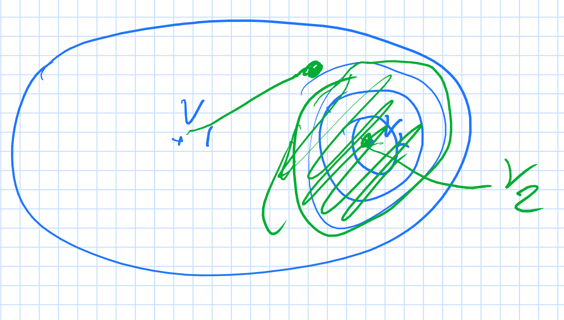
Network deployment

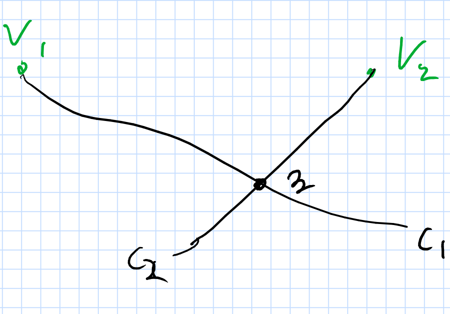
n vertices

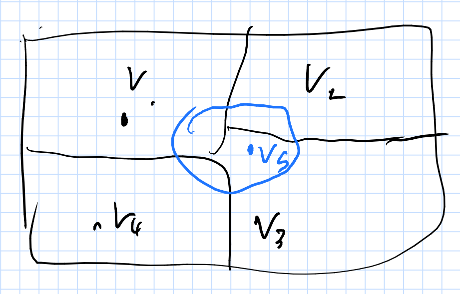
m edges

Compute ordering that min # of messages sent.





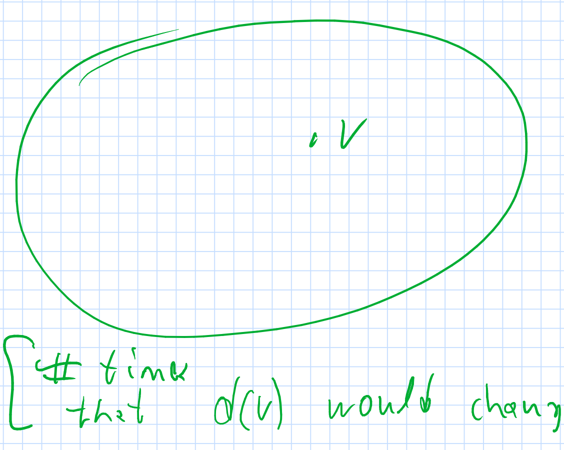




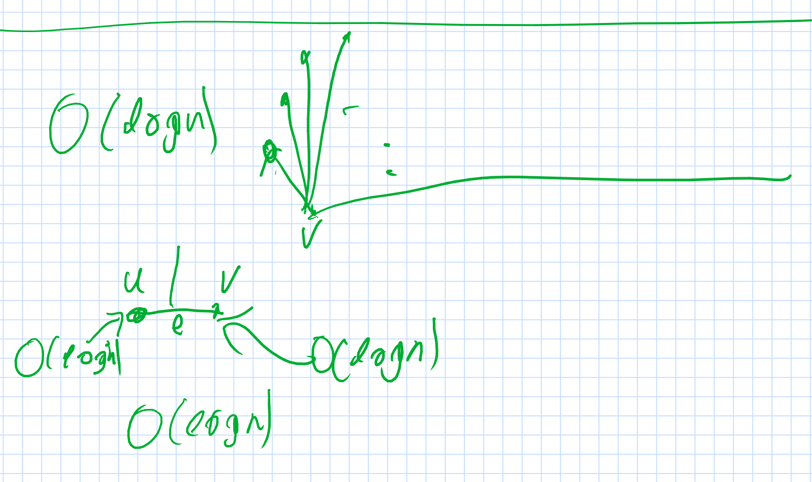

```
relax(e = u→v)
  if (d(v) < d(u) + w(u→v))
     d(v) ← d(u) + w(u+v)
     insert v to the queue / decrease key
       of Dijkstra
```

```
Dijkstra_lazy(s):
  H ← empty heap
  relax all edges coming out of s
  while H is not empty do
    u ← extract_min(H)
    relax_all_edges_from(u)
```

$E\left[\sum_{i=1}^{n} O\left(n_s \log n_s + \#\text{edges adjacent to the vertex}\right)\right] = O\left(n \log n \cdot \log n + m \log n\right)$



E[# time that $d(v)$ would change]

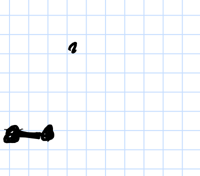


Closest Pair

$P \subseteq \mathbb{R}^2$ n points in the plane.

$P = \{p_1, p_2, \ldots, p_n\}$

$cp(P) = \min_{i<j} \|p_i - p_j\|$



Thm (Rabin, 1976)

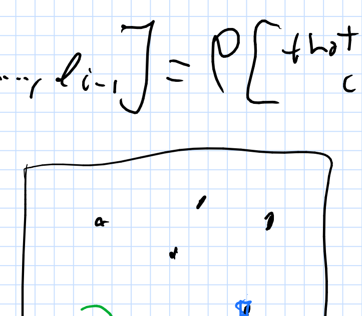
$cp(P)$ can be computed in expected $O(n)$ time.

RIC:

$p_1, p_2, \ldots, p_n$: random permutation of the points of P.

$\ell_i = cp(p_1, p_2, \ldots p_i)$

$P[\ell_i < \ell_1, \ell_2, \ldots, \ell_{i-1}] = P$[that cp distance changed in ith iteration]



$= \frac{2}{i}$

P: set of points, r distance

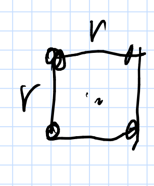
Alg that decides if

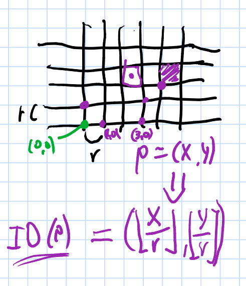
$cp(P) > r$

$= r$

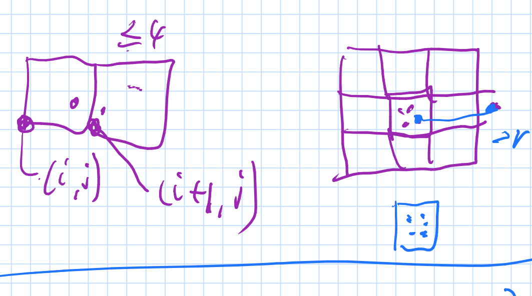
$< r$

Verifier



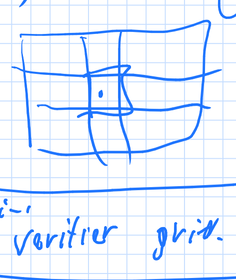


$ID(p) = \left(\left\lfloor \frac{x}{r} \right\rfloor, \left\lfloor \frac{y}{r} \right\rfloor\right)$



```
For i=3,...,n do
  Vi ← verifier(p1,...,pi-1, li-1)   ← O(i)
  insert pi into Vi
  if li < li-1 then update
    current cp distance
```

$O(n)$



→ Rebuild verifier iff $\ell_i < \ell_{i-1}$

otherwise insert p_i into verifier grid.

$\sum_{i=1}^{n} \frac{2}{i} = O(\log n)$ $\quad O(i)$ rebuild

$O(1)$

$O(n \log n)$

E[RT in the ith iteration]

$= P[\ell_i < \ell_{i-1}] O(i)$

$+ P[\ell_i = \ell_{i-1}] \cdot O(1)$

$= \frac{2}{i} O(i) + O(1) = O(1)$

$\Rightarrow E[RT] = O(n)$.