Improved Bounds for Online Preemptive Matching

Danny Segev

with Leah Epstein, Asaf Levin, and Oren Weimann

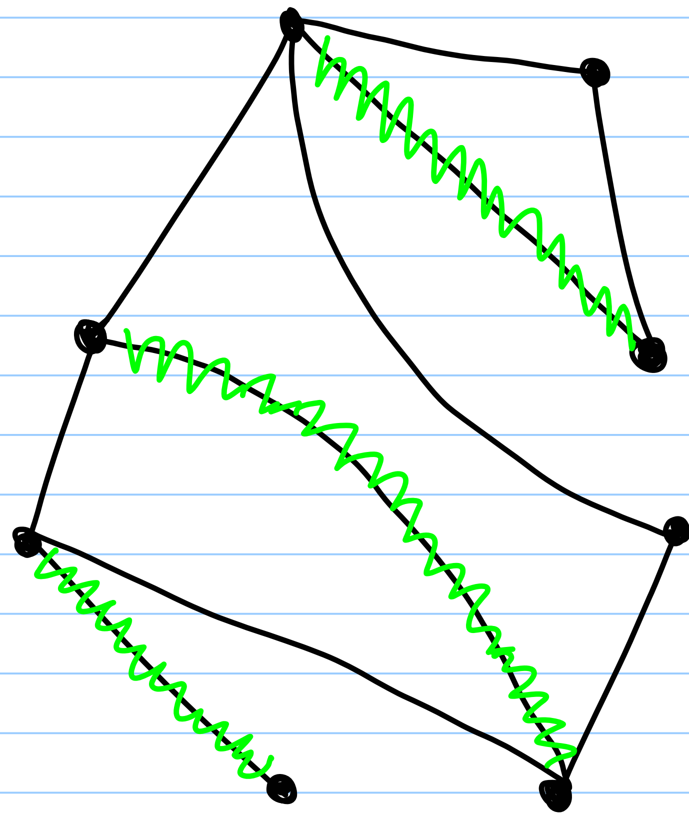
OUTLINE

- Introduction
- Lower bound for max-cardinality matching
- Additional results

INTRODUCTION

MAX-CARDINALITY MATCHING : OFFLINE VERSION

$G=(V,E)$

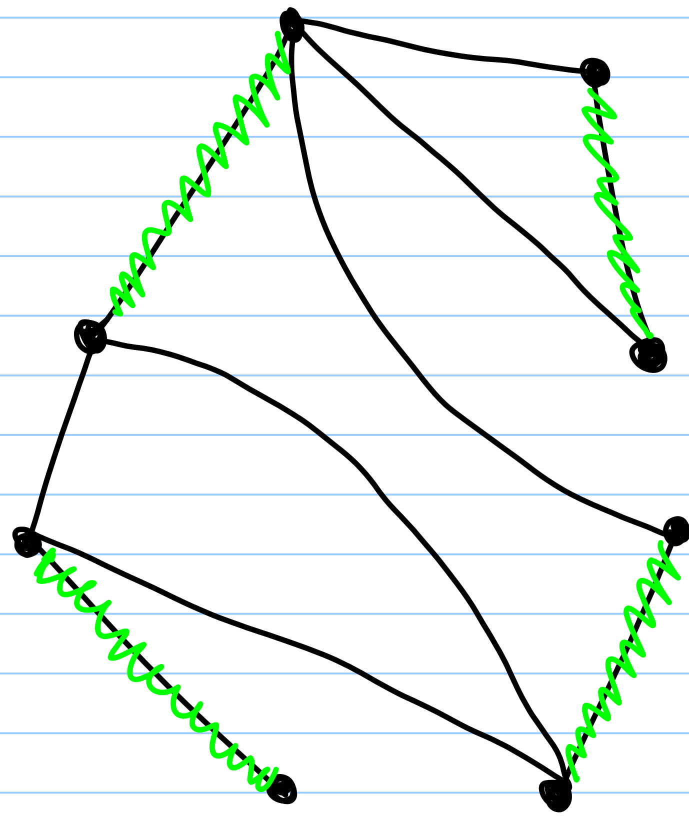


matching: collection of disjoint edges

Objective: Find a matching $M \subseteq E$ of maximum cardinality

MAX-CARDINALITY MATCHING : OFFLINE VERSION

$G=(V,E)$

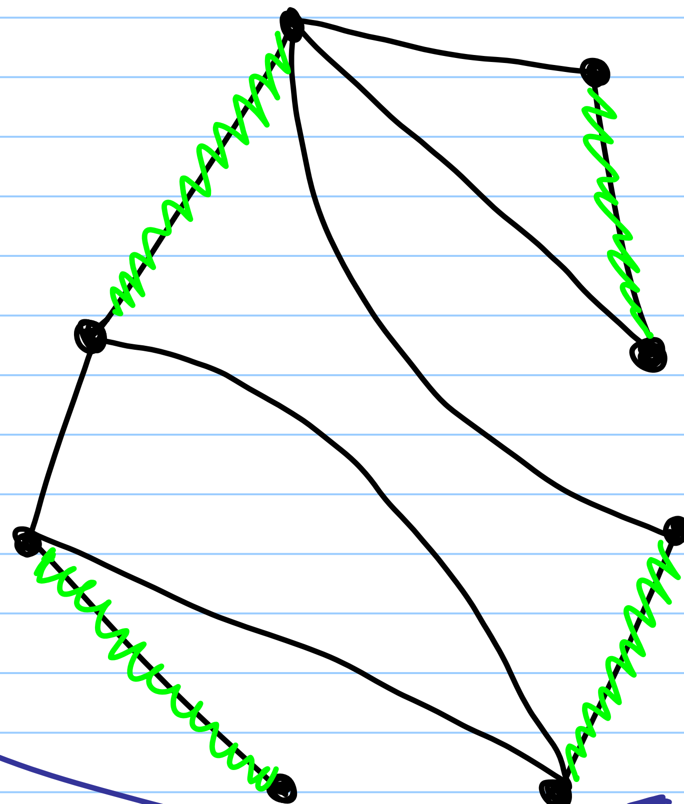


matching: collection of disjoint edges

Objective: Find a matching $M \subseteq E$ of maximum cardinality

MAX-CARDINALITY MATCHING : OFFLINE VERSION

$G=(V,E)$

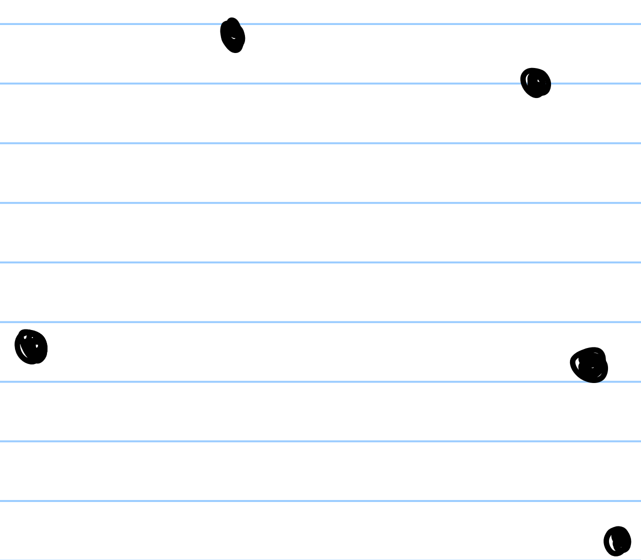


matching: collection of disjoint edges

EFFICIENT ALGORITHMS
+ MASSIVE LITERATURE

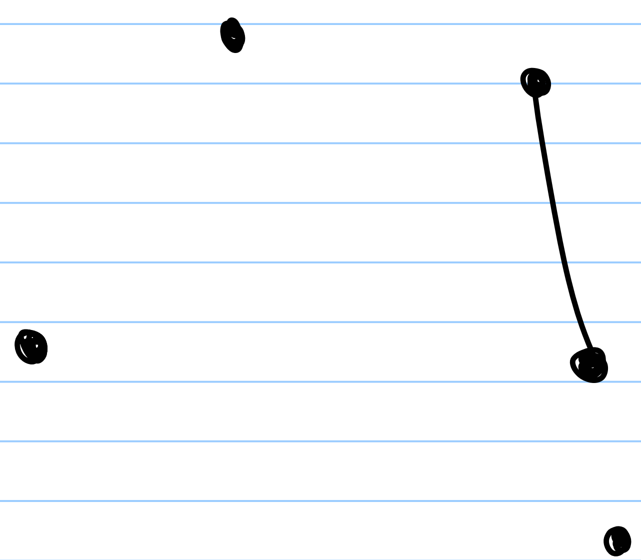
Objective: Find a **matching** $M \subseteq E$ of **maximum cardinality**

MAX-CARDINALITY MATCHING : ONLINE VERSION



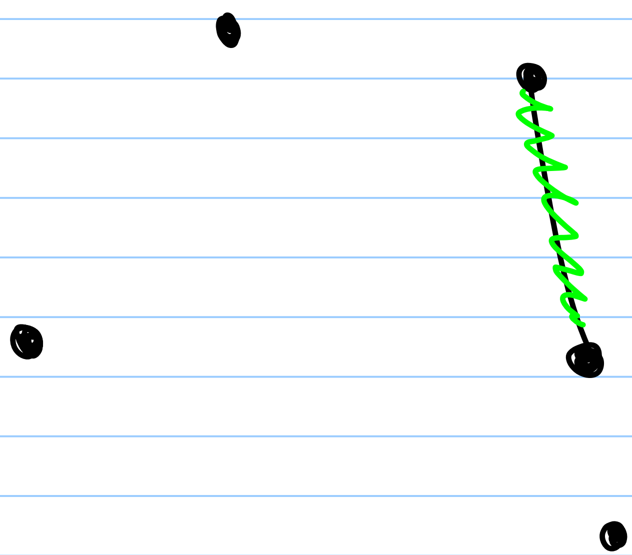
- Edges are revealed one by one
- Algorithm's decision : add the current edge or not ???
- Needs to keep a feasible matching at any time

MAX-CARDINALITY MATCHING : ONLINE VERSION



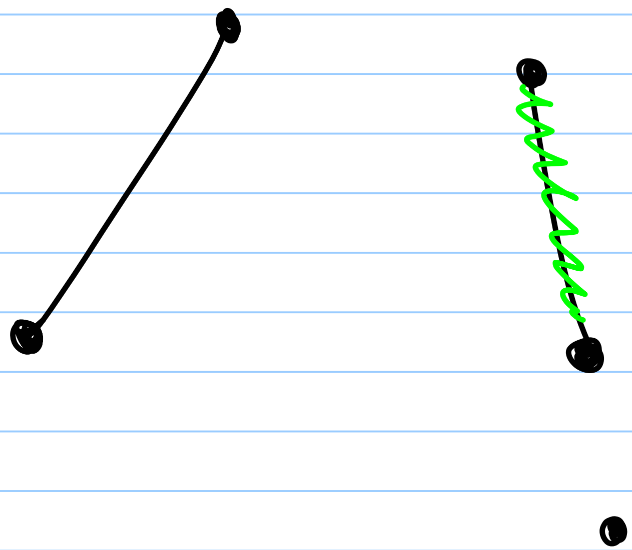
- Edges are revealed one by one
- Algorithm's decision : add the current edge or not ???
- Needs to keep a feasible matching at any time

MAX-CARDINALITY MATCHING : ONLINE VERSION



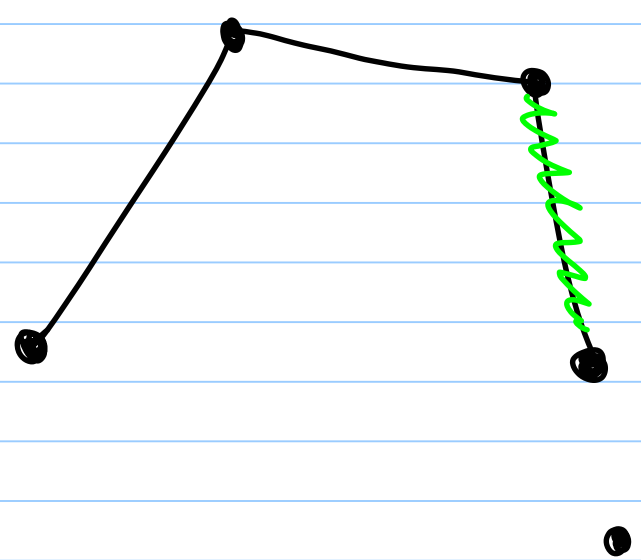
- Edges are revealed one by one
- Algorithm's decision : add the current edge or not ???
- Needs to keep a feasible matching at any time

MAX-CARDINALITY MATCHING : ONLINE VERSION



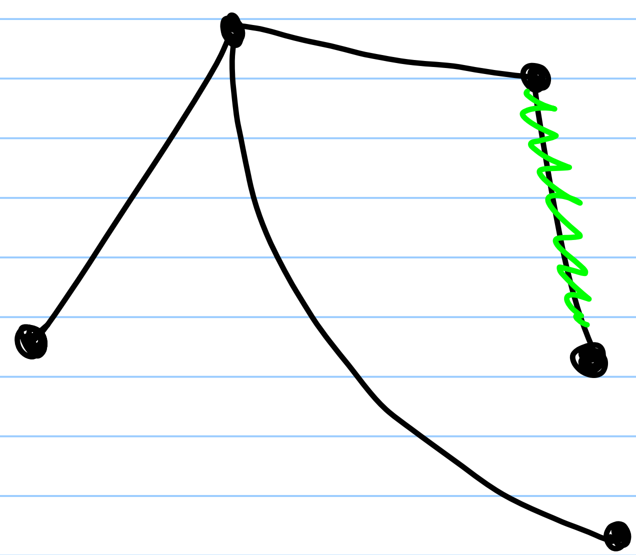
- Edges are revealed one by one
- Algorithm's decision : add the current edge or not ???
- Needs to keep a feasible matching at any time

MAX-CARDINALITY MATCHING : ONLINE VERSION



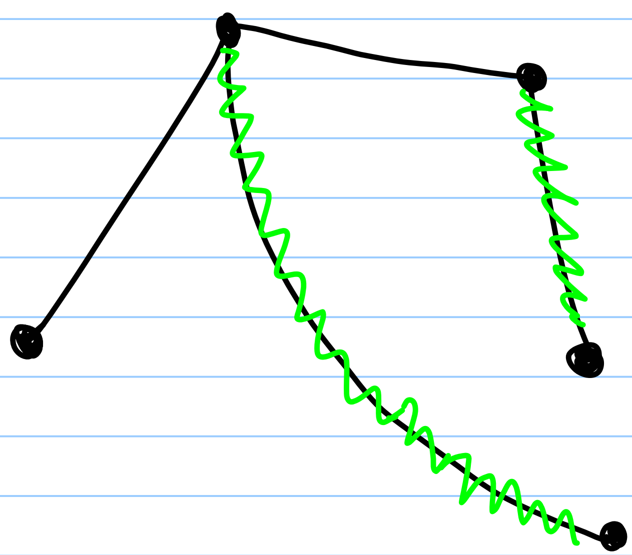
- Edges are revealed one by one
- Algorithm's decision : add the current edge or not ???
- Needs to keep a feasible matching at any time

MAX-CARDINALITY MATCHING : ONLINE VERSION



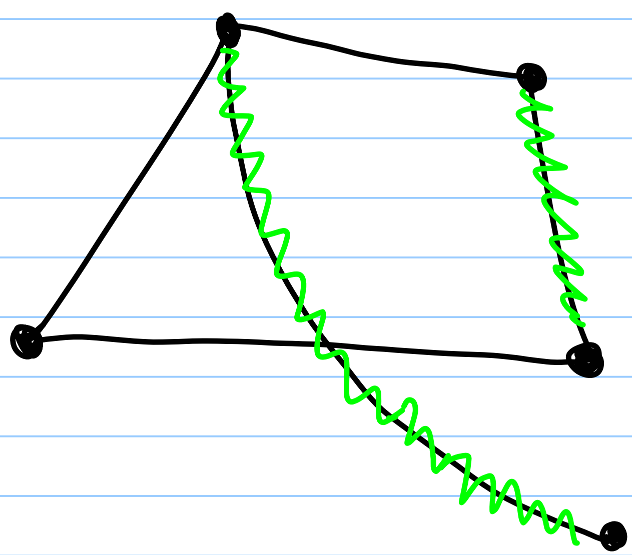
- Edges are revealed one by one
- Algorithm's decision : add the current edge or not ???
- Needs to keep a feasible matching at any time

MAX-CARDINALITY MATCHING : ONLINE VERSION



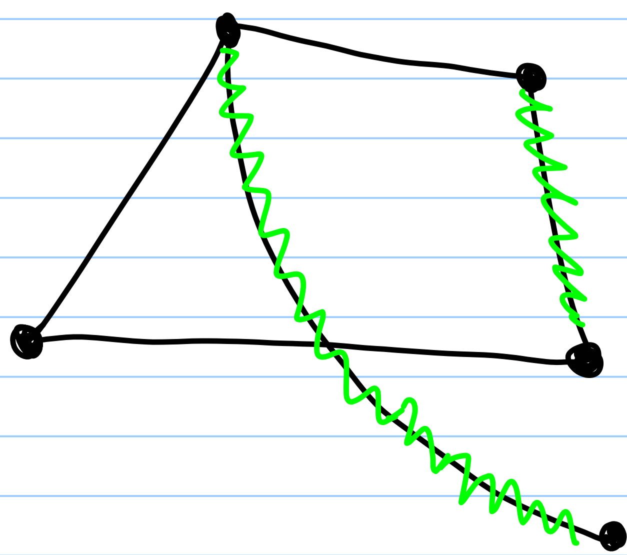
- Edges are revealed one by one
- Algorithm's decision : add the current edge or not ???
- Needs to keep a feasible matching at any time

MAX-CARDINALITY MATCHING : ONLINE VERSION



- Edges are revealed one by one
- Algorithm's decision : add the current edge or not ???
- Needs to keep a feasible matching at any time

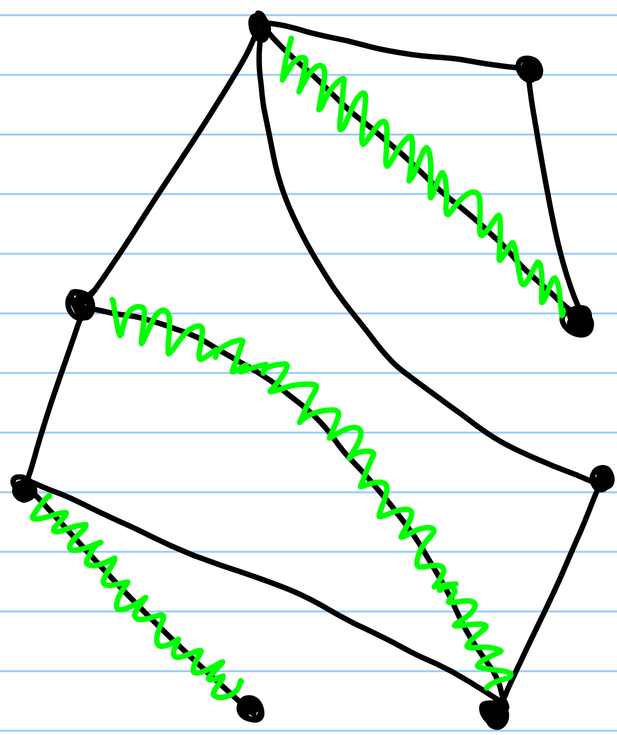
MAX-CARDINALITY MATCHING : ONLINE VERSION



- Preemptive setting: can discard previously-picked edges
- Non-preemptive setting: picked edges are permanent

THIS TALK

COMPETITIVE RATIO



worst possible performance over all input sequences:

$$CR(A) = \sup_{s} \frac{\text{max-matching}(s)}{A\text{-matching}(s)}$$

PREEMPTIVE SETTING: PREVIOUS WORK

- Deterministic algorithms
  - greedy is 2-competitive
  - this is best-possible

- Randomized algorithms
  - greedy has not been beaten yet
  - lower bound of $e/(e-1) \approx 1.581$

    [Karp, Vazirani, and Vazirani '90]

PREEMPTIVE SETTING: PREVIOUS WORK

- Deterministic algorithms
  - greedy is 2-competitive
  - this is best-possible

NEW RESULT: lower bound of $1 + \ln 2 \approx 1.693$

- Randomized algorithms
  - greedy has not been beaten yet
  - lower bound of $e/(e-1) \approx 1.581$

[Karp, Vazirani, and Vazirani '90]

SKETCH OF THE LOWER BOUND PROOF

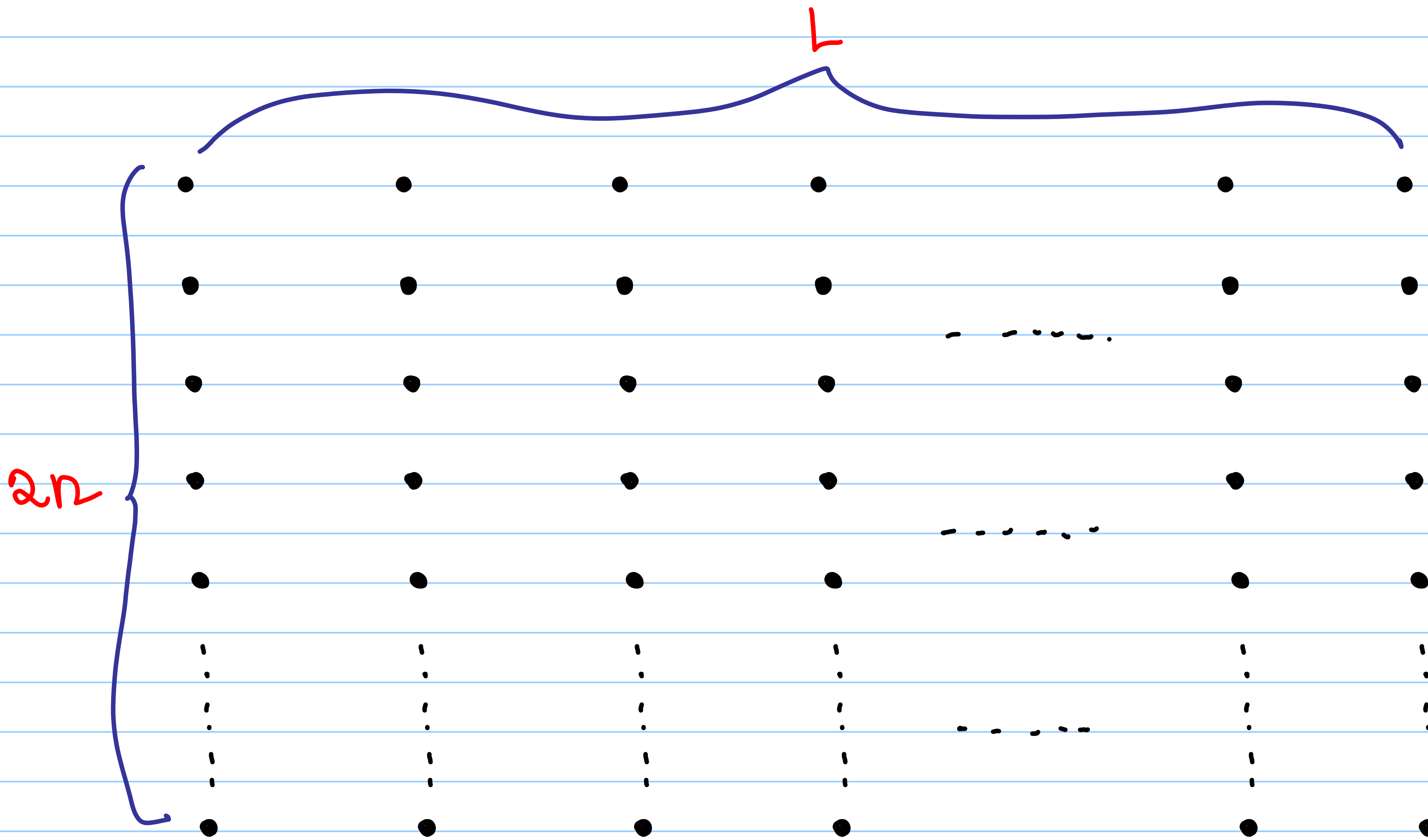
LB FOR RANDOMIZED ALGORITHMS ?

- Online version of Yao's [1977] Lemma:

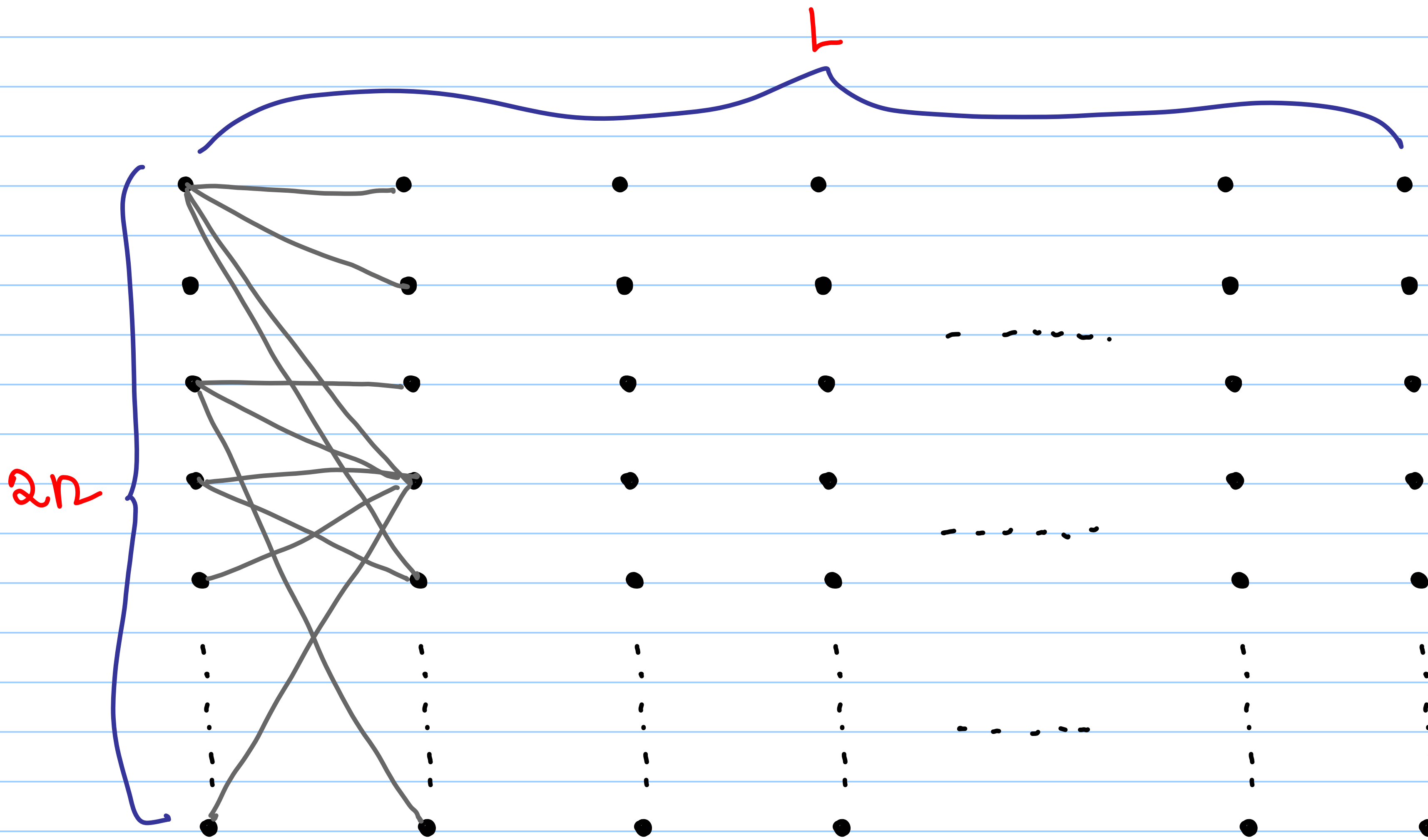
To obtain a lower bound for randomized algorithms, will evaluate the performance of any deterministic algorithm on some probability distribution of the inputs.

- Will assume that:
  - matching ALG is deterministic
  - sequence of revealed edges is random

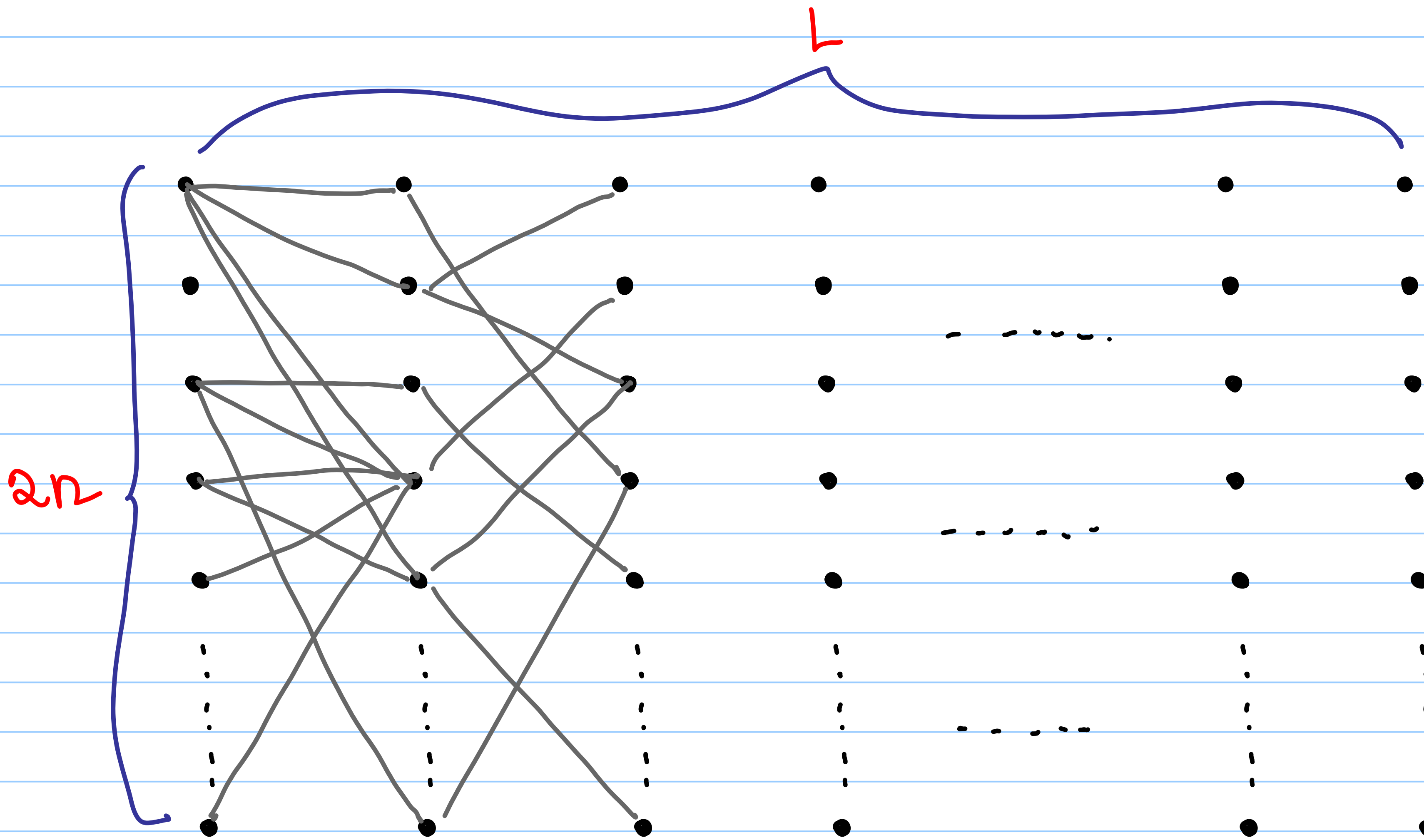
GENERAL STRUCTURE



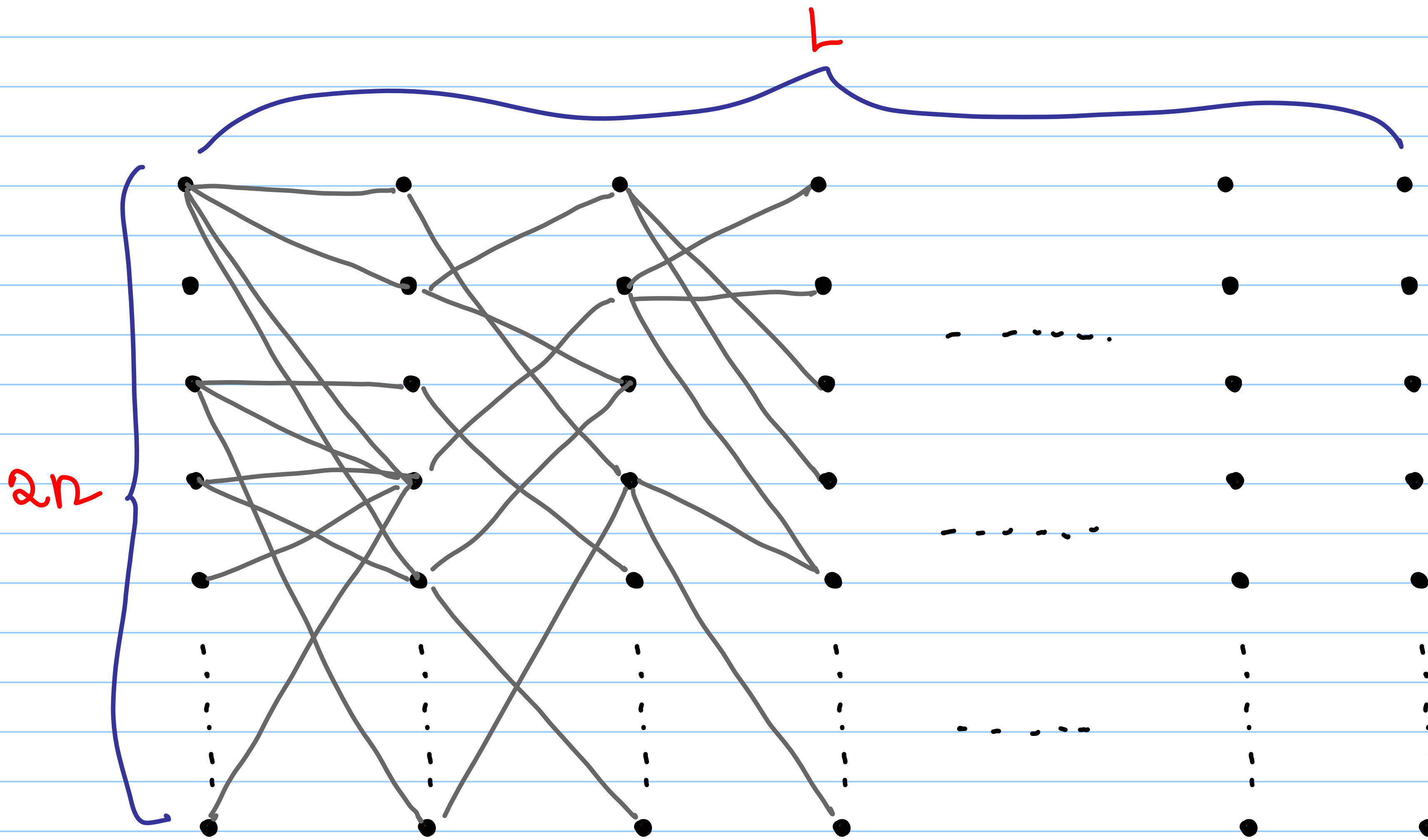
GENERAL STRUCTURE



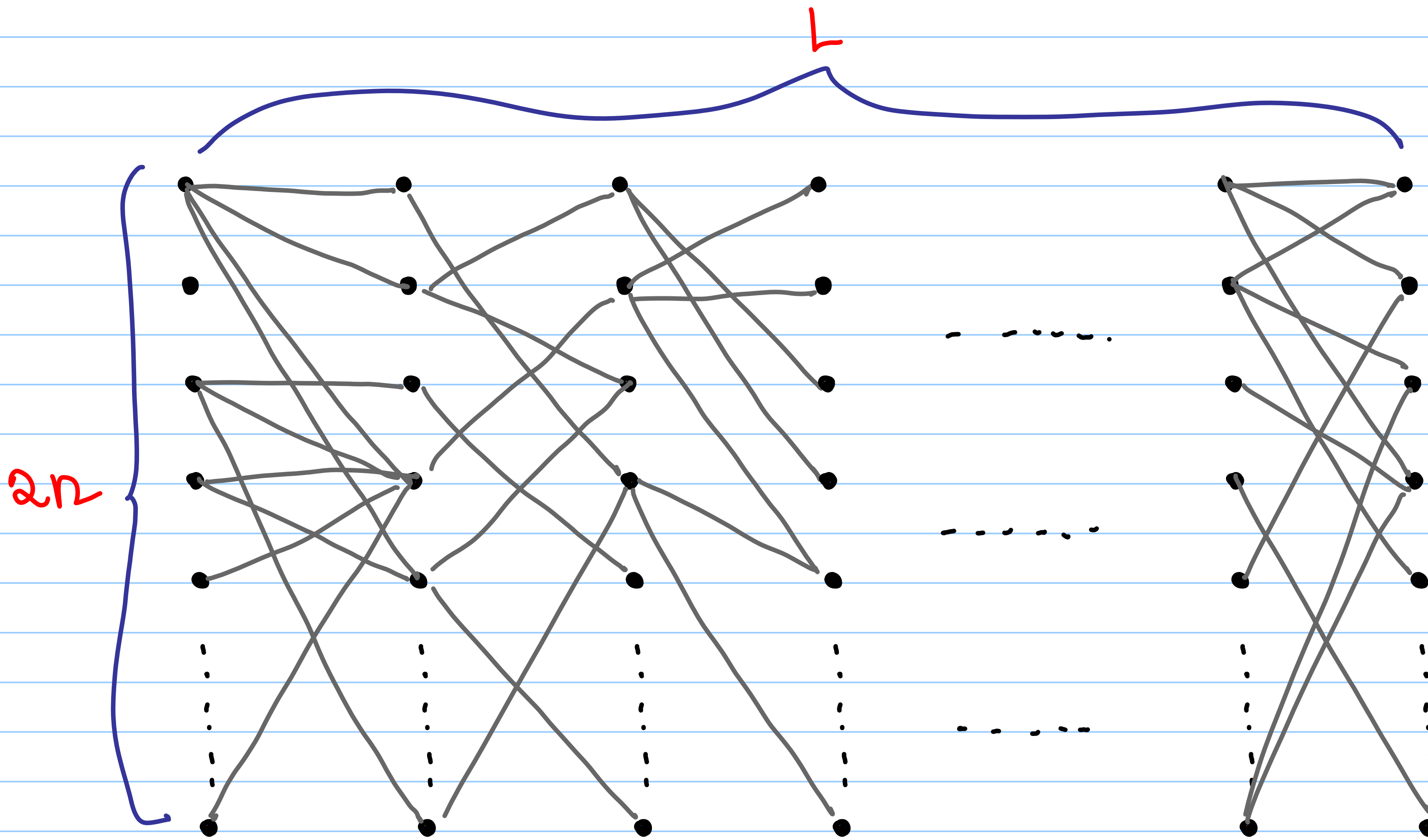
GENERAL STRUCTURE



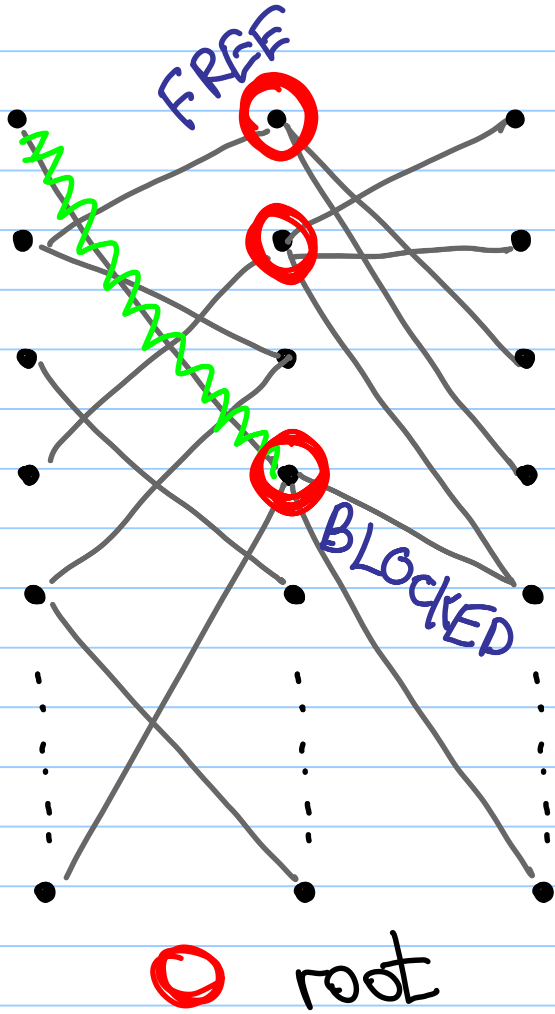
GENERAL STRUCTURE



GENERAL STRUCTURE



HOW DOES ALG OPERATE?



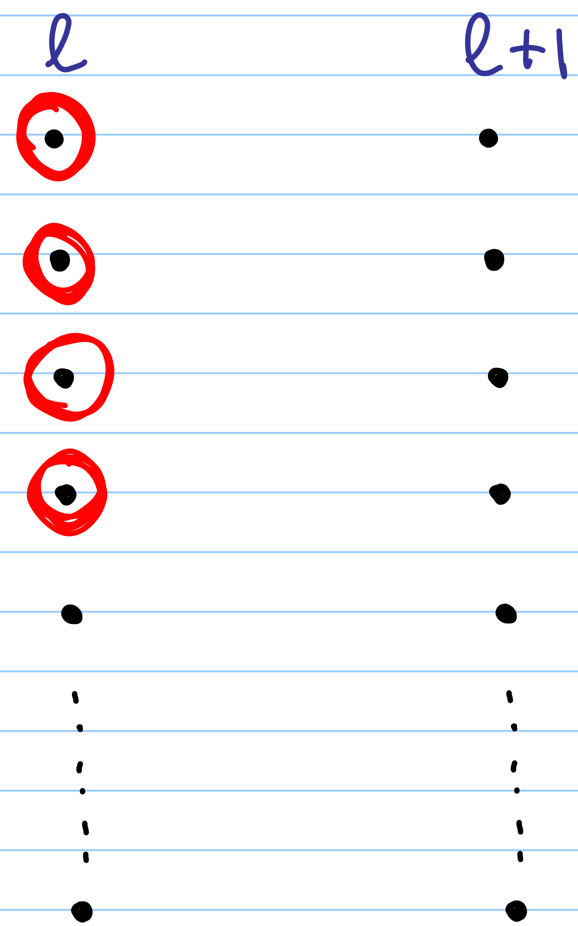
- Edges are revealed only next to roots (always n)
- Blocked roots – not matched
- Free roots – matched (always possible)

$$\Downarrow$$

$$E[ALG] = \sum_{\ell=1}^{L-1} E[F_\ell]$$

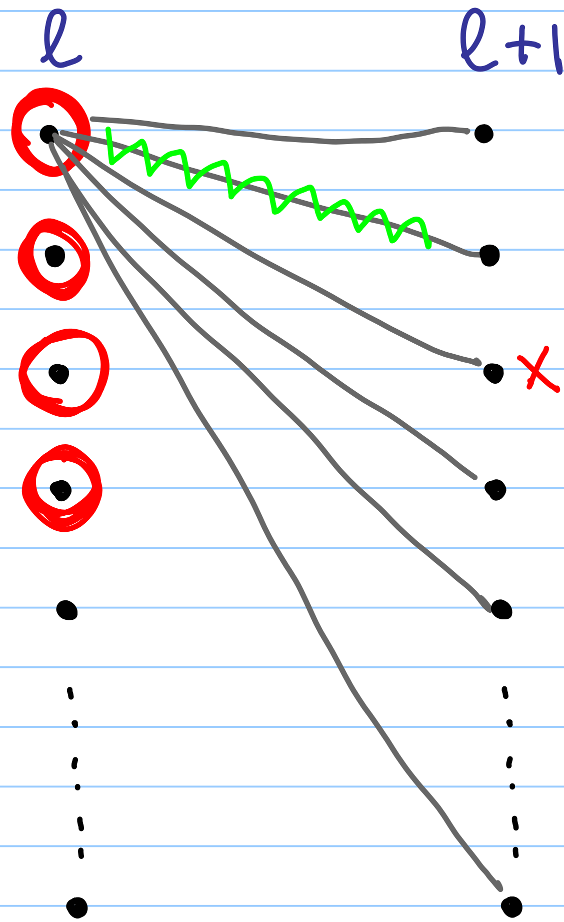
number of free roots in layer ℓ

REVEALING EDGES



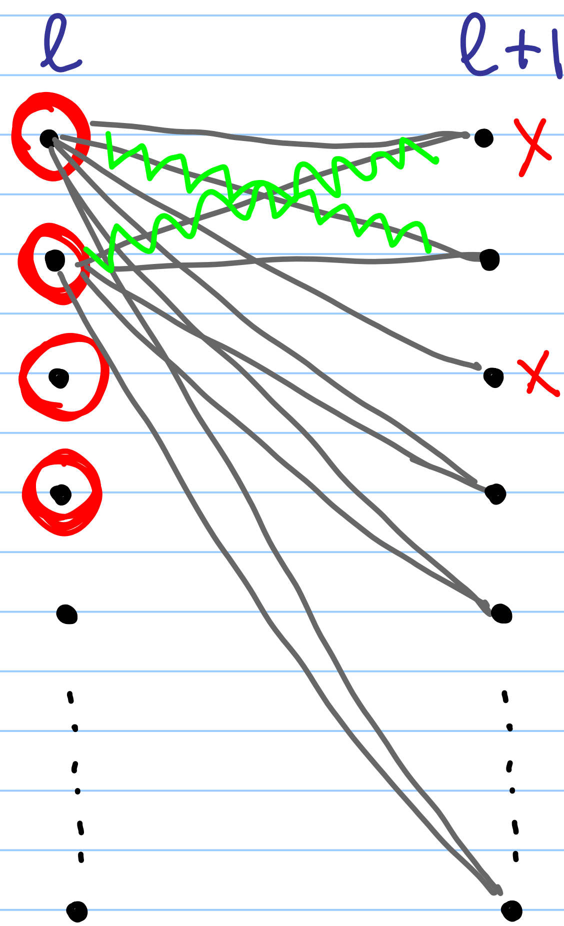
- Pick random permutation of the roots
- Connect current root to all undeleted vertices
- Delete a random vertex
- Roots of layer $\ell+1$: n undeleted vertices

REVEALING EDGES



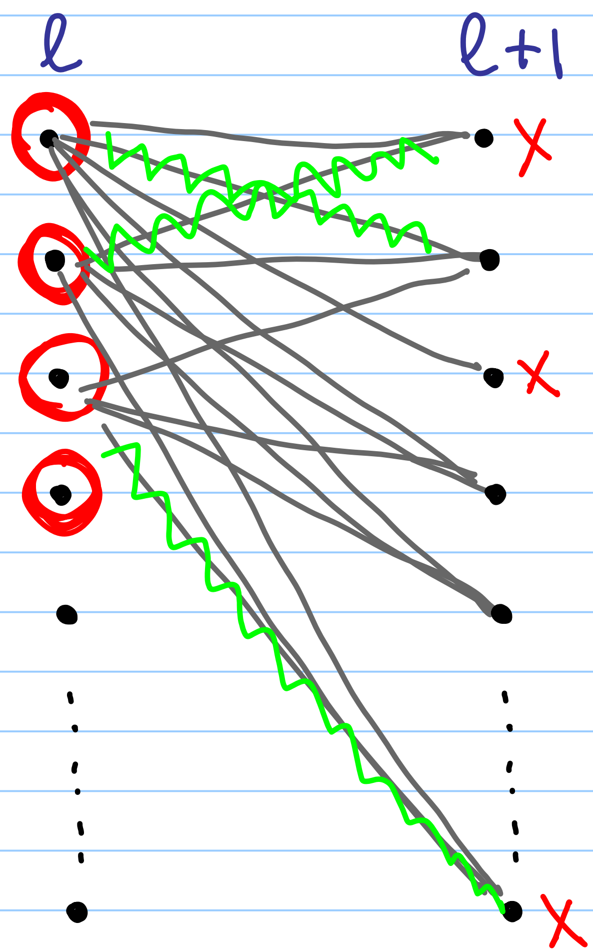
- Pick random permutation of the roots
- Connect current root to all undeleted vertices
- Delete a random vertex
- Roots of layer $\ell+1$: n undeleted vertices

REVEALING EDGES



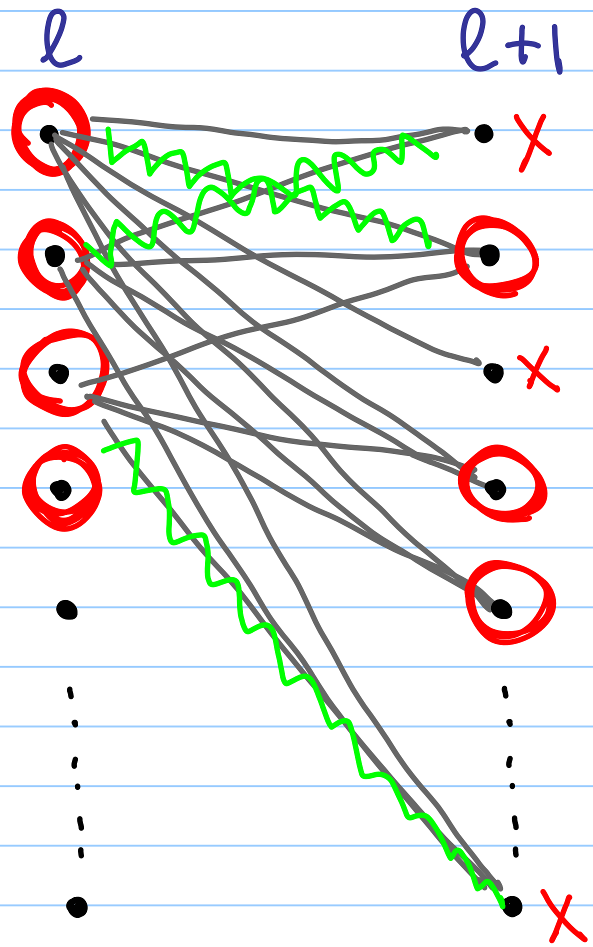
- Pick random permutation of the roots
- Connect current root to all undeleted vertices
- Delete a random vertex
- Roots of layer $\ell+1$: n undeleted vertices

REVEALING EDGES



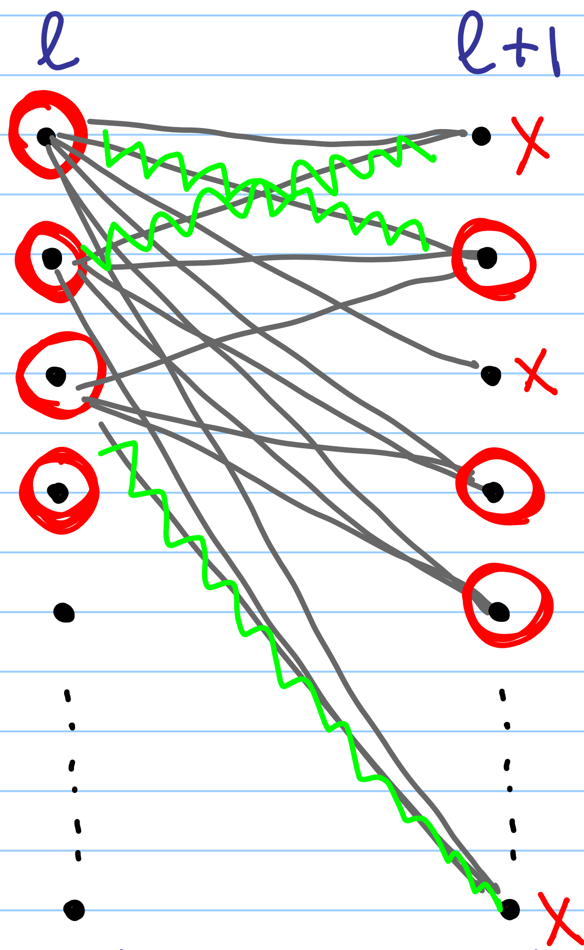
- Pick random permutation of the roots
- Connect current root to all undeleted vertices
- Delete a random vertex
- Roots of layer ℓ+1 : n undeleted vertices

REVEALING EDGES

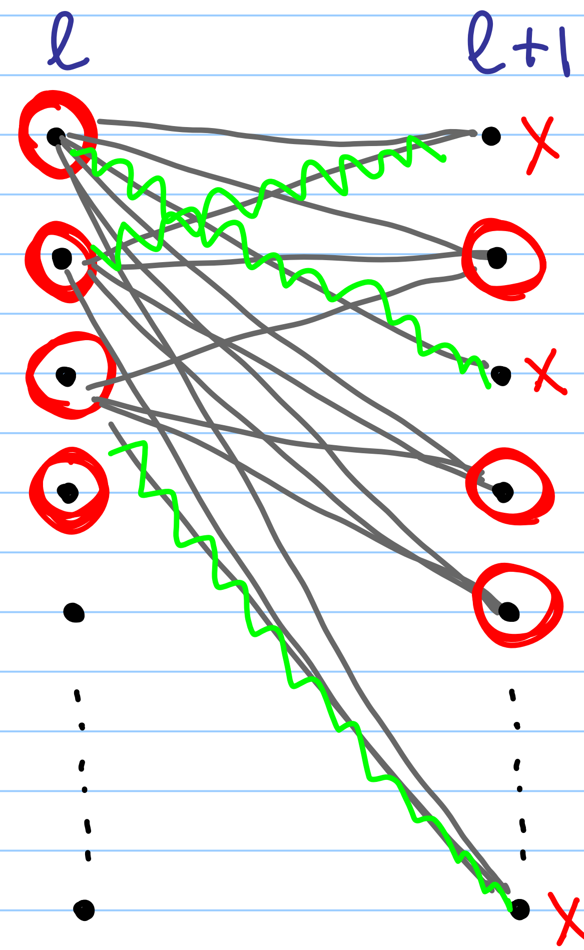


- Pick random permutation of the roots
- Connect current root to all undeleted vertices
- Delete a random vertex
- Roots of layer $\ell+1$: n undeleted vertices

OBSERVATIONS



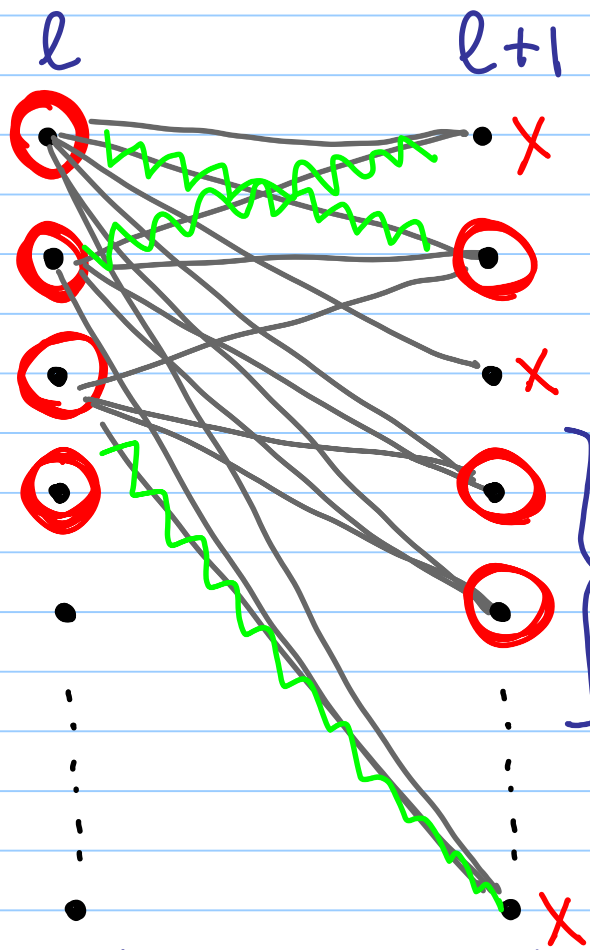
ALG's choices may lead to blocked roots in layer $\ell+1$



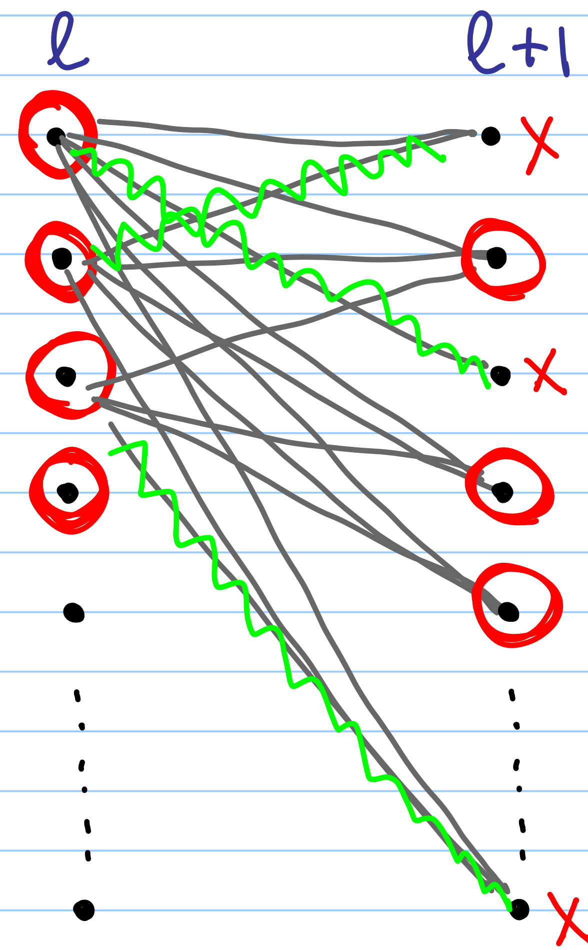
OPT can avoid this

OPT $= (L-1)n$

OBSERVATIONS



$E[F_\ell] = ?$

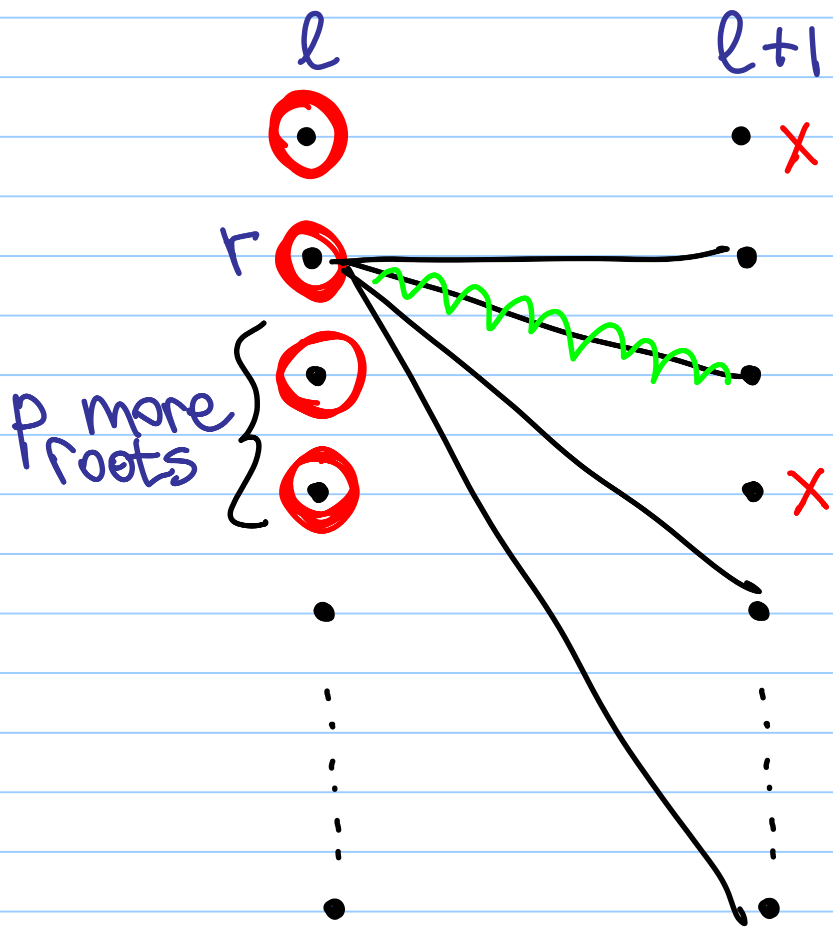


ALG's choices may lead to blocked roots in layer $\ell+1$

OPT can avoid this

$OPT = (L-1)n$

PROBABILITY OF CREATING A BLOCKED ROOT



$$\Pr\left[\begin{matrix}r \text{ creates} \\ \text{blocked root}\end{matrix} \;\middle|\; \begin{matrix}p \text{ more} \\ \text{roots}\end{matrix}\right]$$

$$= \left(1 - \frac{1}{n+p}\right)\left(1 - \frac{1}{n+p-1}\right)\cdots\left(1 - \frac{1}{n+1}\right)$$

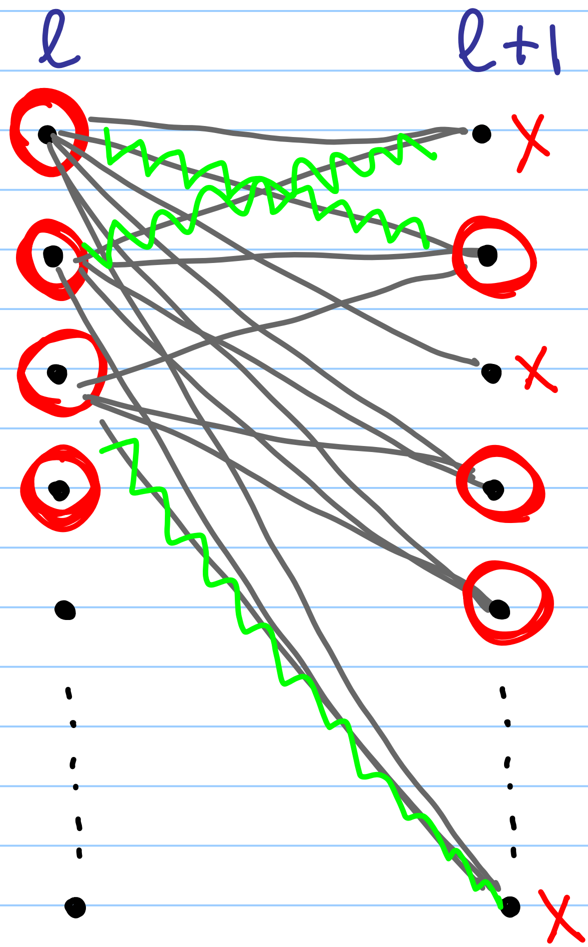
$$= \frac{n}{n+p}$$

$$\Pr\left[\begin{matrix}r \text{ creates} \\ \text{blocked root}\end{matrix}\right] = \sum_{p=0}^{n-1} \frac{1}{n} \cdot \frac{n}{n+p}$$

$$= \ln 2 + \Theta\left(\frac{1}{n}\right)$$

random permutation

COMPUTING $E[F_\ell]$



$$E[F_{\ell+1}] = n - E[B_{\ell+1}]$$

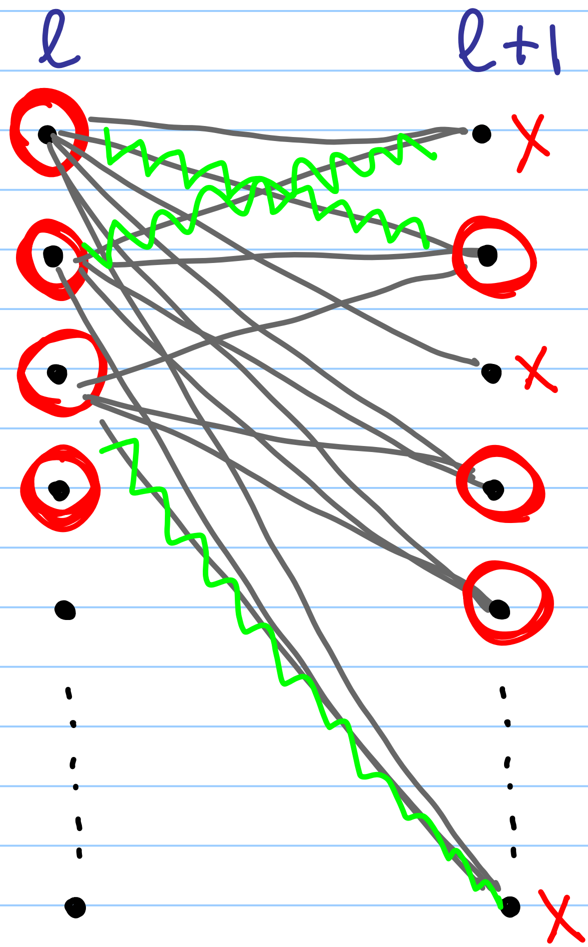
$$= n - E[E[B_{\ell+1} \mid F_\ell]]$$

$$= n - \left(\ln 2 + \theta\left(\tfrac{1}{n}\right)\right) \cdot E[F_\ell]$$

$$\Downarrow$$

$$E[F_\ell] = \frac{n}{1+\ln 2} + \theta(n \cdot \ln^\ell 2)$$

RESULTING LOWER BOUND



$$E[ALG] = \sum_{\ell=1}^{L-1} E[F_\ell]$$

$$= \frac{1}{1+\ln 2}(L-1)n + \Theta(n)$$

$$OPT = (L-1)n$$

$$\Downarrow$$

$$\frac{OPT}{E[ALG]} \xrightarrow[L,n\to\infty]{} 1+\ln 2$$

ADDITIONAL RESULTS

UPPER BOUND FOR WEIGHTED GRAPHS

- An upper bound of $3+2\sqrt{2} \approx 5.828$ can be attained by a deterministic algorithm [Feigenbaum et al. '05] [McGregor '05]
- This bound is best-possible without randomization [Varadaraja '11]

UPPER BOUND FOR WEIGHTED GRAPHS

- An upper bound of $3+2\sqrt{2} \approx 5.828$ can be attained by a deterministic algorithm [Feigenbaum et al. '05] [McGregor '05]
- This bound is best-possible without randomization [Varadaraja '11]

NEW RESULT: upper bound of ≈ 5.356 via a randomized algorithm

UPPER BOUND FOR WEIGHTED GRAPHS

- An upper bound of $3+2\sqrt{2} \approx 5.828$ can be attained by a deterministic algorithm [Feigenbaum et al. '05] [McGregor '05]
- This bound is best-possible without randomization [Varadaraja '11]

NEW RESULT: upper bound of ≈ 5.356 via a randomized algorithm

THANK YOU !