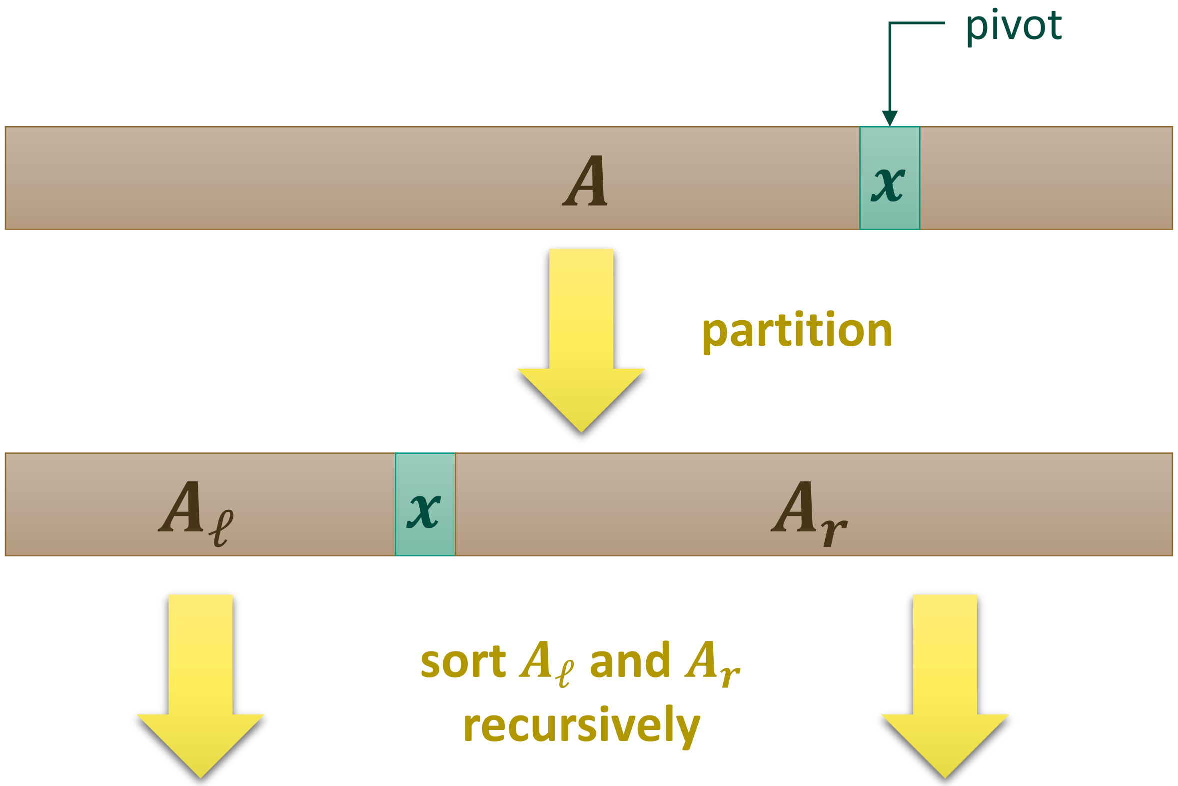
universität freiburg

Algorithm Theory – WS 2024/25

Chapter 7 : Randomization II
Randomized Quicksort / Randomized Min Cut Algorithm

Fabian Kuhn
Dept. of Computer Science
Algorithms and Complexity

Randomized Quicksort



Randomized Quicksort:

pick pivot uniformly at random

Randomized Quicksort Analysis

Randomized Quicksort: pick uniform random element as pivot

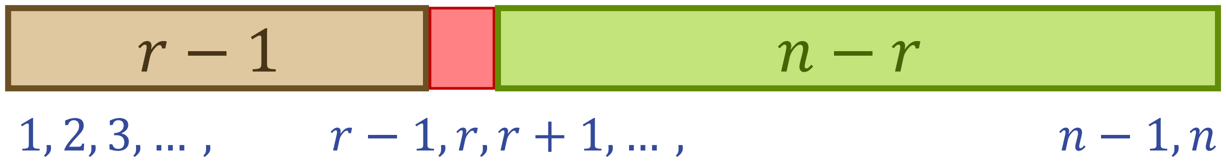
Running Time of sorting n elements:

- Let us just count the number of comparisons
- In the partitioning step, all $n-1$ non-pivot elements have to be compared to the pivot
- **Number of comparisons:**

depends on choice of pivot

$$\boldsymbol{n-1\ +\ \#\text{comparisons in recursive calls}}$$

- **If rank of pivot is $\boldsymbol{r}$:** recursive calls with $\boldsymbol{r-1}$ and $\boldsymbol{n-r}$ elements



Randomized Quicksort Analysis

We have seen that:

$= {}^{1}/_{n}$

$$\overset{T(n)}{\mathbb{E}[C]} = \sum_{r=1}^{n} \mathbb{P}(R = r) \cdot \left(n - 1 + \overbrace{\mathbb{E}[C_\ell | R = r]}^{T(r-1)} + \overbrace{\mathbb{E}[C_r | R = r]}^{T(n-r)}\right)$$

Define: $\boldsymbol{T(n)}$: expected number of comparisons when sorting n elements

$$\mathbb{E}[C] = T(n)$$
$$\mathbb{E}[C_\ell | R = r] = T(r-1)$$
$$\mathbb{E}[C_r | R = r] = T(n-r)$$

Recursion:

$$\boldsymbol{T(n) = \sum_{r=1}^{n} \frac{1}{n} \cdot \big(n - 1 + T(r-1) + T(n-r)\big)}$$

$$\boldsymbol{T(0) = T(1) = 0}$$

Randomized Quicksort Analysis

$i = r - 1$

Theorem: The expected number of comparisons when sorting n elements using randomized quicksort is $T(n) \leq 2n \ln n$.

Proof: *(by induction on n)*

$$T(n) = \sum_{r=1}^{n} \frac{1}{n} \cdot \big(n - 1 + T(r-1) + T(n-r)\big), \qquad T(0) = T(1) = 0$$

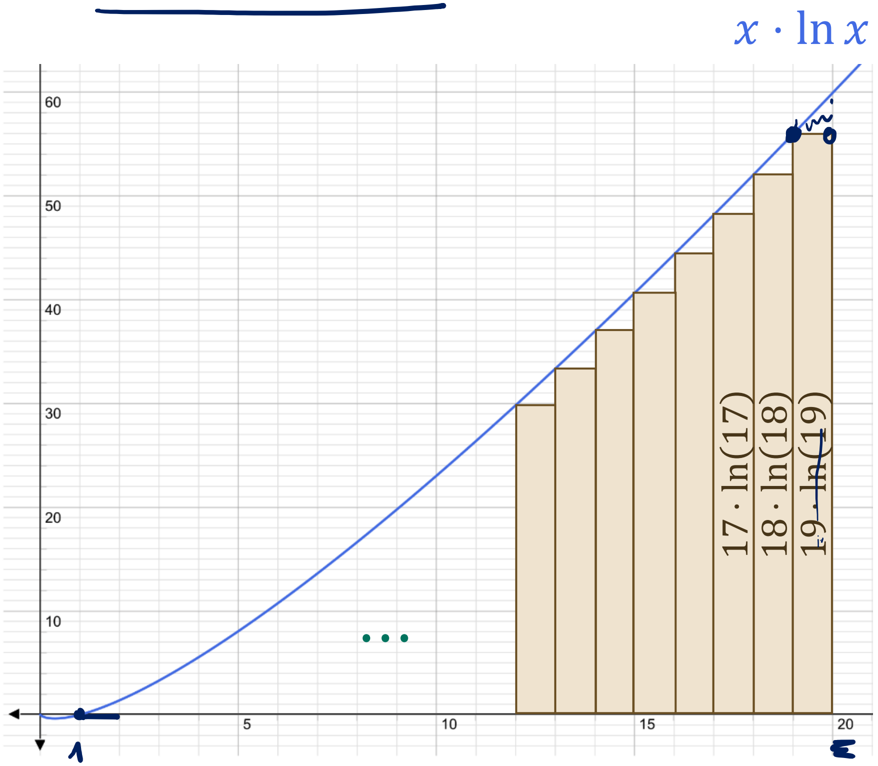
$$= n - 1 + \frac{1}{n} \cdot \sum_{i=0}^{n-1} \big(T(i) + T(n-i-1)\big)$$

$$= n - 1 + \frac{2}{n} \cdot \sum_{i=1}^{n-1} T(i)$$

$$\leq n - 1 + \frac{4}{n} \cdot \sum_{i=1}^{n-1} i \cdot \ln i$$

$$< n - 1 + \frac{4}{n} \cdot \int_{1}^{n} x \ln x \, dx$$

induction hypothesis



Randomized Quicksort Analysis

Theorem: The expected number of comparisons when sorting n elements using randomized quicksort is $T(n) \leq 2n \ln n$.

Proof:

$$T(n) \leq n - 1 + \frac{4}{n} \cdot \int_1^n x \ln x \, dx$$

$$T(n) \leq n - 1 + \frac{4}{n} \cdot \left[\frac{n^2 \ln n}{2} - \frac{n^2}{4} + \frac{1}{4}\right]$$

$$= \cancel{n} - 1 + 2n \ln n - \cancel{n} + \frac{1}{n}$$

$$= 2n \ln n + \underbrace{\left(\frac{1}{n} - 1\right)}_{<0}$$

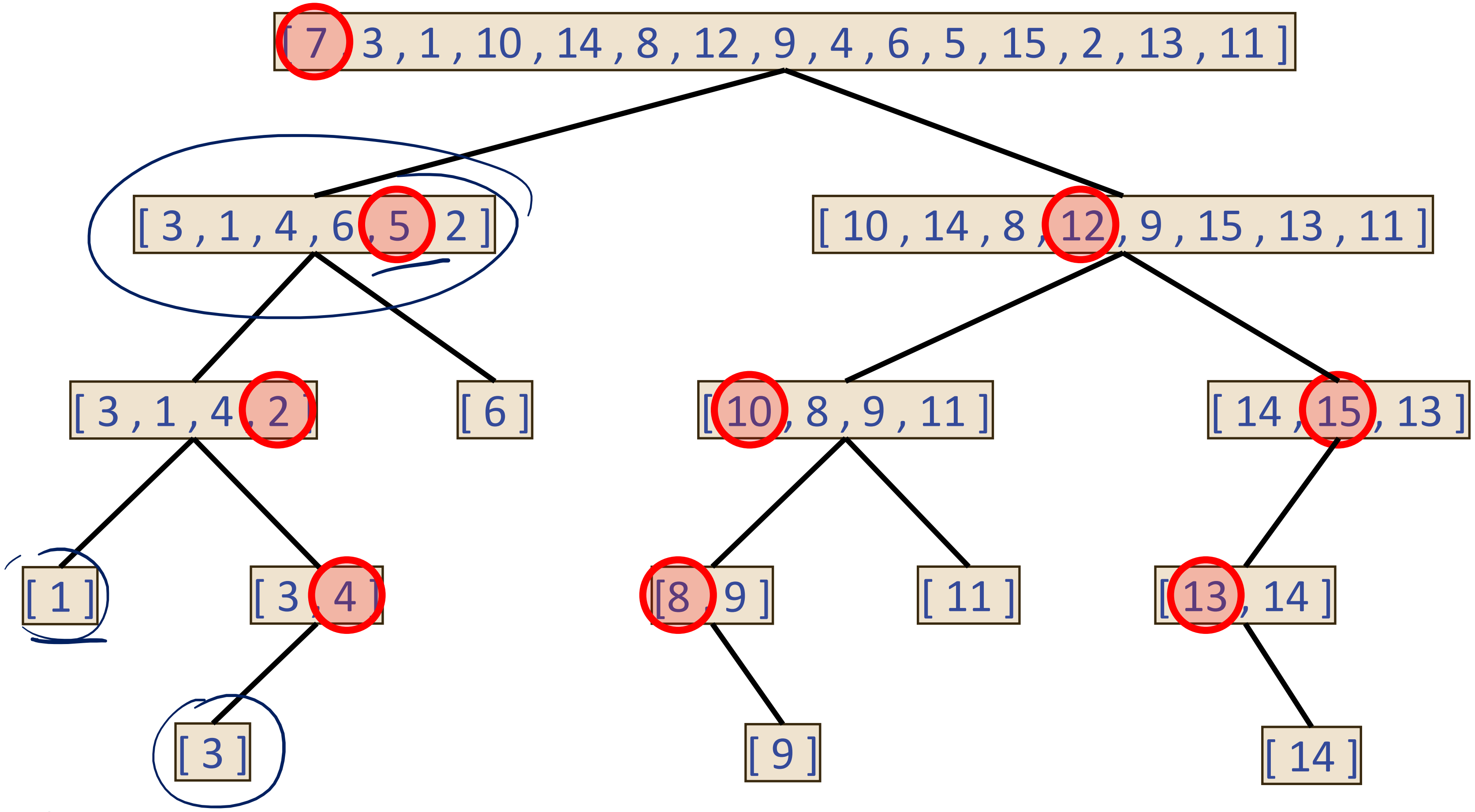
$$< 2n \ln n$$

$$\int x \ln x \, dx = \frac{x^2 \ln x}{2} - \frac{x^2}{4}$$

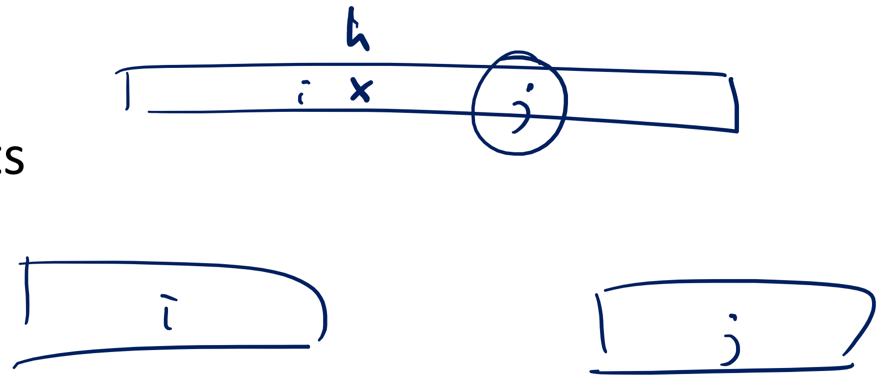
Alternative Randomized Quicksort Analysis

Array to sort: [7 , 3 , 1 , 10 , 14 , 8 , 12 , 9 , 4 , 6 , 5 , 15 , 2 , 13 , 11]

Viewing quicksort run as a tree:



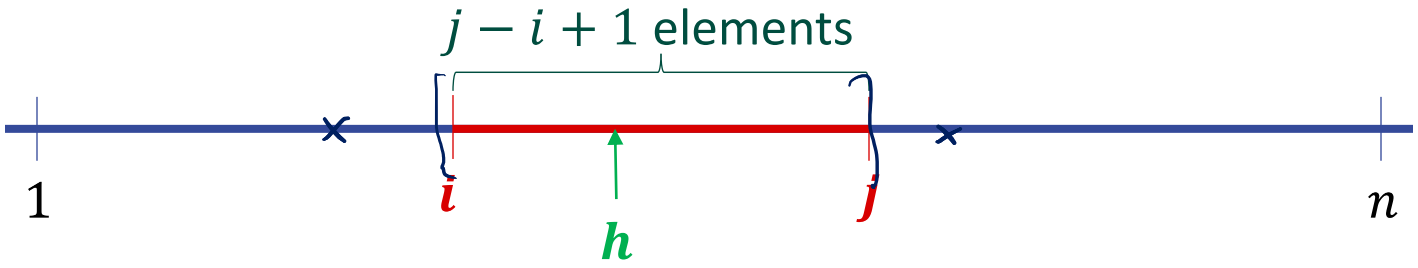
Comparisons



- Comparisons are only between pivot and non-pivot elements
- Every element can only be the pivot once:
  → every 2 elements can only be compared once!

- W.l.o.g., assume that the elements to sort are $1,2,\dots,n$
- Elements i and j are compared if and only if either i or j is a pivot before any element $h: i < h < j$ is chosen as pivot
  - i.e., iff i is an ancestor of j or j is an ancestor of i

2 3
i j



$$\mathbb{P}(\text{comparison between } i \text{ and } j) = \frac{2}{j-i+1}$$

Counting Comparisons

Random variable for every pair of elements (i, j), $i < j$:

$$\boldsymbol{X_{ij}} = \begin{cases} 1, & \text{if there is a comparison between } i \text{ and } j \\ 0, & \text{otherwise} \end{cases}$$

$$\mathbb{P}(X_{ij} = 1) = \frac{2}{j - i + 1}, \qquad \mathbb{E}[X_{ij}] = \frac{2}{j - i + 1}$$

Number of comparisons: $\boldsymbol{X}$

$$X = \sum_{i<j} X_{ij}$$

- What is $\mathbb{E}[X]$?

Randomized Quicksort Analysis

Theorem: The expected number of comparisons when sorting n elements using randomized quicksort is $T(n) \leq 2n \ln n$.

Proof:

- **Linearity of expectation:**
  For all random variables $X_1, \dots, X_n$ and all $a_1, \dots, a_n \in \mathbb{R}$,

$$\mathbb{E}\left[\sum_{i}^{n} a_i X_i\right] = \sum_{i}^{n} a_i \mathbb{E}[X_i].$$

$$X = \sum_{i<j} X_{ij} \quad \Longrightarrow \quad \mathbb{E}[X] = \mathbb{E}\left[\sum_{i<j} X_{ij}\right] = \sum_{i<j} \mathbb{E}[X_{ij}]$$

$$= \sum_{i<j} \frac{2}{j-i+1} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1}$$

Randomized Quicksort Analysis

Theorem: The expected number of comparisons when sorting n elements using randomized quicksort is $T(n) \leq 2n \ln n$.

Proof:

$$\mathbb{E}[X] = \underline{2 \cdot \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{j-i+1}} = 2 \cdot \sum_{i=1}^{n-1} \sum_{\ell=2}^{n-i+1} \frac{1}{\ell}$$

$$\leq 2 \cdot \sum_{i=1}^{n-1} \underbrace{\sum_{\ell=2}^{n} \frac{1}{\ell}}_{= H(n) - 1}$$

$$= 2 \cdot (n-1) \cdot (H(n) - 1)$$

$$\leq 2 \cdot (n-1) \cdot \ln n$$

Harmonic Series:

$$H(k) := \sum_{i=1}^{k} \frac{1}{i}$$

$$H(k) \leq 1 + \ln k$$

Quicksort: High Probability Bound

- We have seen that the number of comparisons of randomized quicksort is $O(n \log n)$ in expectation.

- Can we also show that the number of comparisons is $O(n \log n)$ with high probability?

- **Recall:**

  On each recursion level, each pivot is compared once with each other element that is still in the same "part"

Counting Number of Comparisons

- We looked at 2 ways to count the number of comparisons
  - recursive characterization of the expected number
  - number of different pairs of values that are compared

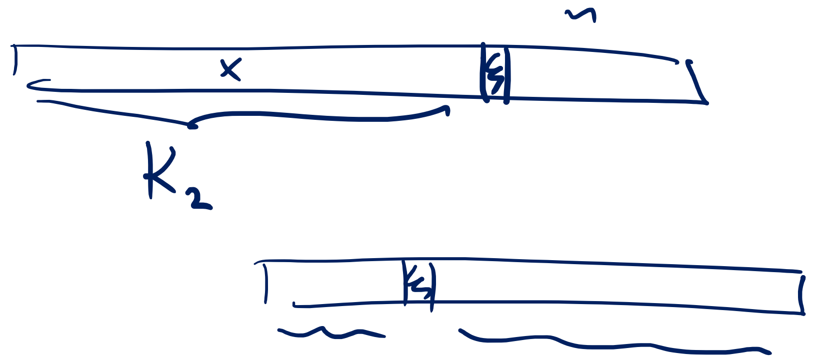
Let's consider yet another way:

- Each comparison is between a pivot and a non-pivot
- How many times is a specific array element x compared as a non-pivot?

Element x is compared as a non-pivot to a pivot once in every recursion level until one of the following two conditions apply:

x is chosen as a pivot or x is alone

Successful Recursion Level



- Consider a specific recursion level ℓ
  - Where the first recursion level is level 1

Define K_ℓ as follows: focus on elem. x

- If x is contained in a subarray on recursion level ℓ, then K_ℓ is defined as the length of the subarray containing x on level ℓ.
  - We therefore have $K_1 = n$ and $K_{\ell+1} \leq K_\ell$ for all $\ell \geq 1$
- If x has been chosen as a pivot before level ℓ, we set $K_\ell := 1$

#comparisons of x as non-pivot $\leq$ #levels ℓ for which $K_\ell > 1$

Definition: We say that recursion level ℓ is successful for element x iff the following is true:

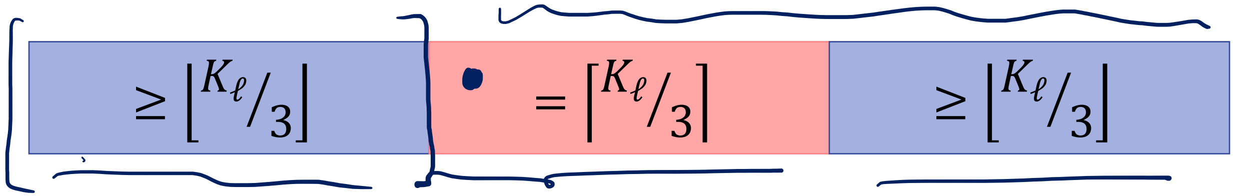
$$K_{\ell+1} = 1 \quad \text{or} \quad K_{\ell+1} \leq \frac{2}{3} \cdot K_\ell$$

Successful Recursion Level

Lemma: For every recursion level ℓ and every array element x, it holds that level ℓ is successful for x with probability at least ${}^{1}/_{3}$, independently of what happens in other recursion levels.

Proof:

- Assume that $K_\ell > 1$, otherwise level ℓ is trivially successful



- If pivot is in the middle part, both remaining parts have size

$$\leq K_\ell - \left\lfloor {}^{K_\ell}/_{3} \right\rfloor - 1 \leq {}^{2}/_{3} \cdot K_\ell.$$

  - In this case, level ℓ is successful

- The probability that the pivot in in the middle part is $\geq {}^{1}/_{3}$.

Number of Successful Recursion Levels

Lemma: If among the first ℓ recursion levels, at least $\log_{3/2}(n)$ are successful for element x, we have $K_{\ell+1} = 1$.

Proof:

- We know that

$$K_1 = n, \qquad \forall i \geq 1 : K_{i+1} \leq K_i$$

- If level i is successful, then $K_{i+1} \leq 2/3 \cdot K_i$ or $K_{i+1} = 1$

- If s among the first ℓ levels are successful, then

$$K_{\ell+1} \leq \max\left\{1, n \cdot \left(2/3\right)^s\right\}$$

- If $s \geq \log_{3/2}(n)$, then $K_{\ell+1} \leq 1$.

Chernoff Bounds

- Let $X_1, \dots, X_n$ be independent 0-1 random variables and define $p_i \coloneqq \mathbb{P}(X_i = 1)$.
- Consider the random variable $X = \sum_{i=1}^{n} X_i$
- We have $\mu \coloneqq \mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \sum_{i=1}^{n} p_i$

If $p_i = p$ for all i:

$X \sim \mathrm{Bin}(n, p)$

Chernoff Bound (Lower Tail):

$$\forall \boldsymbol{\delta} > \mathbf{0}:\ \mathbb{P}(\boldsymbol{X} < (\mathbf{1} - \boldsymbol{\delta})\boldsymbol{\mu}) < \boldsymbol{e}^{-\boldsymbol{\delta}^2 \boldsymbol{\mu}/2}$$

Chernoff Bound (Upper Tail):

$$\forall \boldsymbol{\delta} > \mathbf{0}:\ \mathbb{P}(\boldsymbol{X} > (\mathbf{1} + \boldsymbol{\delta})\boldsymbol{\mu}) < \left(\frac{\boldsymbol{e}^{\boldsymbol{\delta}}}{(\mathbf{1} + \boldsymbol{\delta})^{\mathbf{1}+\boldsymbol{\delta}}}\right)^{\boldsymbol{\mu}} < \boldsymbol{e}^{-\boldsymbol{\delta}^2 \boldsymbol{\mu}/3}$$

holds for $\boldsymbol{\delta} \le \mathbf{1}$

Chernoff Bounds, Example

$$p_i = p = \frac{1}{2}, \qquad \mu := \mathbb{E}[X] = np = \frac{n}{2}$$

Assume that a fair coin is flipped n times. What is the probability to have

1. less than $n/3$ heads?

$$\mathbb{P}\left(X < \frac{n}{3}\right) = \mathbb{P}\left(X < \left(1 - \frac{1}{3}\right) \cdot \frac{n}{2}\right) < e^{-\frac{1}{2} \cdot \frac{1}{3^2} \cdot \frac{n}{2}} = e^{-n/36}$$

$$\mathbb{P}(X < (1-\delta)\mu) < e^{-\frac{\delta^2}{2}\mu}$$

2. more than $0.51n$ tails?

$$\mathbb{P}\left(X < (1 + 0.02) \cdot \frac{n}{2}\right) < e^{-\frac{0.02^2}{3} \cdot \frac{n}{2}} \approx e^{-0.0000667n}$$

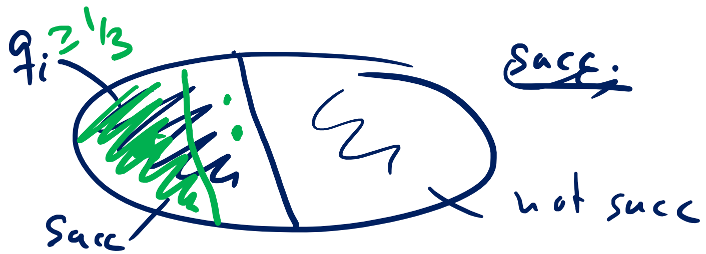
$$\mathbb{P}(X > (1+\delta)\mu) < e^{-\frac{\delta^2}{3}\mu}$$

3. less than $n/2 - \sqrt{c \cdot n \ln n}$ tails?

$$\mathbb{P}\left(X < \left(1 - \frac{2\sqrt{c \cdot n \ln n}}{n}\right) \cdot \frac{n}{2}\right) < e^{-\frac{\cancel{4}c \cdot \cancel{n} \ln n}{\cancel{2}\cancel{n^2}} \cdot \frac{\cancel{n}}{\cancel{2}}} = e^{-c \cdot \ln n} = \frac{1}{n^c}$$

With high probability, #heads/tails $= \frac{n}{2} \pm O\left(\sqrt{n \log n}\right)$

Number of Comparisons for x



Lemma: For every array element x, with high probability, as a non-pivot, x is compared to a pivot at most $O(\log n)$ times.

Proof:

- Consider some level $i \geq 1$, and let if level i not successful

  $1 - q_i$ be the probability

$$q_i := \mathbb{P}(\text{level } i \text{ successful for } x \mid \text{history up to level } i)$$

- Previous lemma $\Longrightarrow q_i \geq {}^1/_3$

  level i not succ $\rightarrow X_i = 0$

  $X_i = 1 \rightarrow$ level i succ.

- Define random variable

$$X_i := \begin{cases} 0 & \text{if level } i \text{ not successful for } x \\ 1 & \text{with probability } \frac{1/3}{q_i} \text{ if level } i \text{ successful for } x \\ 0 & \text{otherwise} \end{cases}$$

- Then, $\mathbb{P}(X_i = 1) = {}^1/_3$ and X_i are independent for different i

$$\mathbb{P}(X_i = 0) = \mathbb{P}(i \text{ succ}) \cdot \frac{1/3}{q_i} = q_i \cdot \frac{1/3}{q_i} = 1/3$$

Number of Comparisons for x

Lemma: For every array element x, with high probability, as a non-pivot, x is compared to a pivot at most $O(\log n)$ times.

Proof:

- X_i independent, $\mathbb{P}(X_i = 1) = {}^1\!/_3$, $X_i = 1 \Longrightarrow$ level i successful
- Consider the first t levels and define $X \coloneqq \sum_i^t X_i$
  - $\mathbb{E}[X] = {}^1\!/_3 \cdot t$
  - $X \leq$ successful levels for x among first t levels
- Hence, if $X \geq \log_{3/2}(n)$, then $K_{t+1} = 1$
- We thus need that for any const. $c > 0$ and some $t = O(\log n)$,

$$\mathbb{P}\left(X < \log_{3/2}(n)\right) \leq \frac{1}{n^c}$$

Number of Comparisons for x

Lemma: For every array element x, with high probability, as a non-pivot, x is compared to a pivot at most $O(\log n)$ times.

Proof:

- $\mu := \mathbb{E}[X] = {}^1/_3 \cdot t$, for $c > 0$ and some $t = O(\log n)$, we need

$$\mathbb{P}\left(X < \log_{3/2}(n)\right) \leq \frac{1}{n^c}$$

$$e^{-\frac{\mu}{8}} \leq n^{-c}$$

- **Chernoff:** $\mathbb{P}(X < (1-\delta)\mu) \leq e^{-\frac{\delta^2}{2}\cdot\mu} \implies \mathbb{P}(X < {}^{\mu}/_2) \leq e^{-\frac{\mu}{8}}$

$$\mathbb{P}(X < \log_{3/2} n) < n^{-c}$$

- We need $\mu \geq 2 \cdot \log_{3/2}(n)$ such that ${}^{\mu}/_2 \geq \log_{3/2}(n)$

$$\frac{\mu}{2} \geq \log_{3/2} n$$

- We need $\mu \geq 8c \cdot \ln n$ such that $e^{-\mu/8} \leq n^{-c}$

- We can therefore choose $t = 3 \cdot \mu = O(\log n)$.

$$t = c' \cdot \log n$$

Number of Comparisons

Theorem: With high probability, the total number of comparisons is at most $\boldsymbol{O(n \log n)}$.

Proof:

- For every const. $c > 0$, there exists const. $\alpha > 0$, s.t. for every element x, the number of comparisons for element x as a non-pivot is $\leq \alpha \ln n$ with probability at least $1 - 1/n^c$.

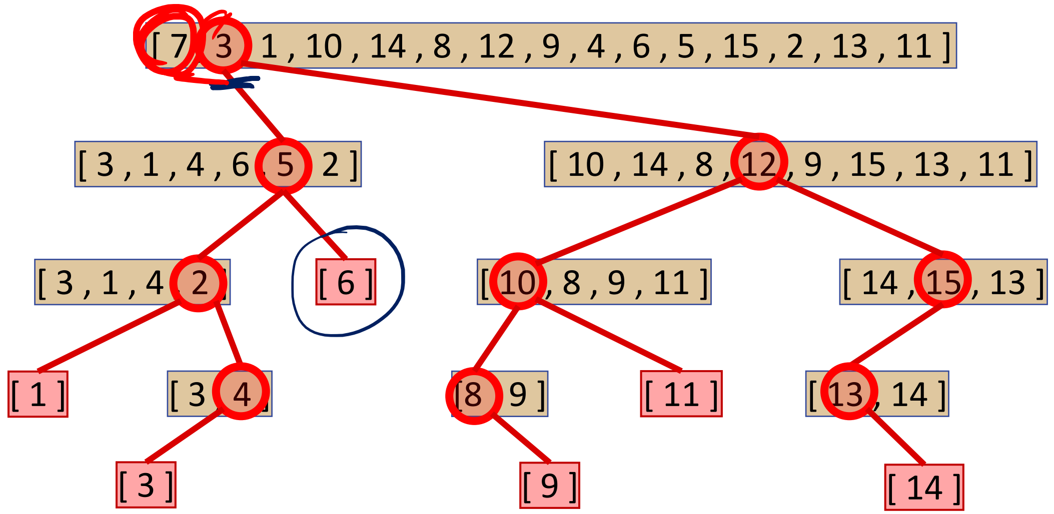
- Define event $\mathcal{E}_x \coloneqq \{\#\text{comparisons for } x \text{ as non–pivot} > \alpha \ln n\}$
  - $\mathbb{P}(\mathcal{E}_x) \leq n^{-c}$

- Union bound over all events $\mathcal{E}_x$:

$$\mathbb{P}\left(\bigcup_{x=1}^{n} \mathcal{E}_x\right) \leq \sum_{x=1}^{n} \mathbb{P}(\mathcal{E}_x) \leq n \cdot \frac{1}{n^c} = \frac{1}{n^{c-1}}$$

Relation to Random Binary Search Trees

Consider Recursion Tree: Label each subarray of size > 1 by the pivot and each subarray of size $= 1$ by the element in it.



- We get a binary search tree (BST) on the n elements
  - Corresponds to the BST with a random insertion order
- #comparisons of element x as non-pivot $=$ depth of x in tree
  - Our analysis shows that the height of a random BST is $O(\log n)$, w.h.p.
- #comp. of rand. quicksort $= n \cdot$ average depth in a random BST

Types of Randomized Algorithms

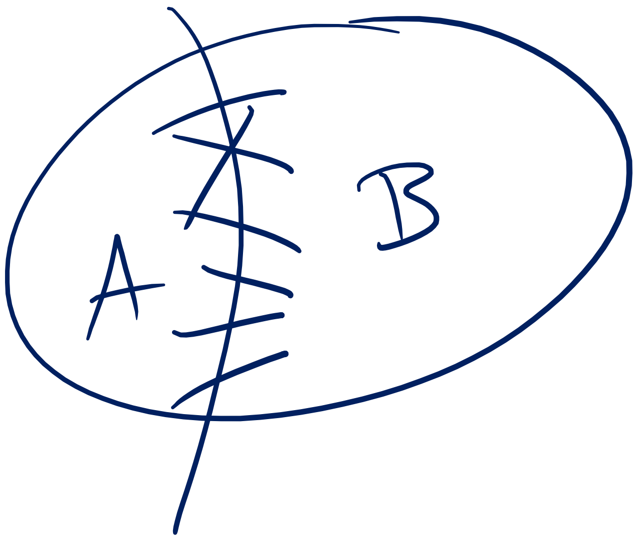
Las Vegas Algorithm:

- always a correct solution
- running time is a random variable
- **Example:** randomized quicksort, contention resolution

Monte Carlo Algorithm:

- probabilistic correctness guarantee (mostly correct)
- fixed (deterministic) running time
- **Example:** primality test

Minimum Cut



Reminder: Given a graph $G = (V, E)$, a cut is a partition (A, B) of V such that $V = A \cup B$, $A \cap B = \emptyset$, $A, B \neq \emptyset$

Size of the cut $(\boldsymbol{A}, \boldsymbol{B})$: # of edges crossing the cut

- For weighted graphs, total edge weight crossing the cut

Goal: Find a cut of minimal size (i.e., of size $\lambda(G)$)

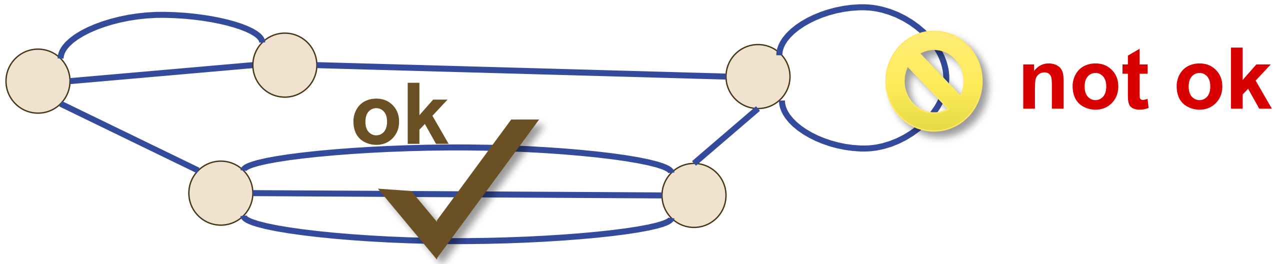
Maximum-flow based algorithm:

- Fix s, compute min s-t-cut for all $t \neq s$
- $O\big(m \cdot \lambda(G)\big) = O(mn)$ per s-t cut
- Gives an $O\big(mn\lambda(G)\big) = O(mn^2)$-algorithm

$O(n^4)$

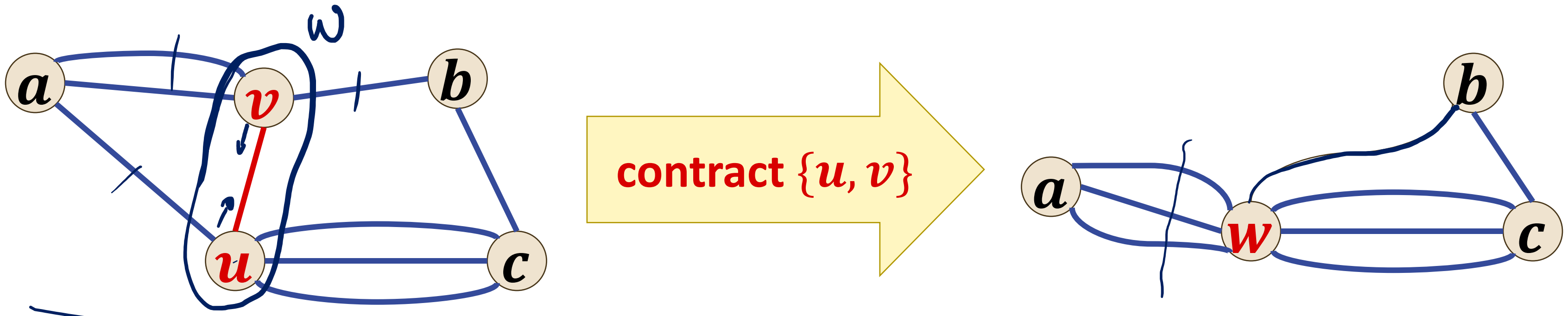
Edge Contractions

- In the following, we consider multi-graphs that can have multiple edges (but no self-loops)



Contracting edge $\{u, v\}$:

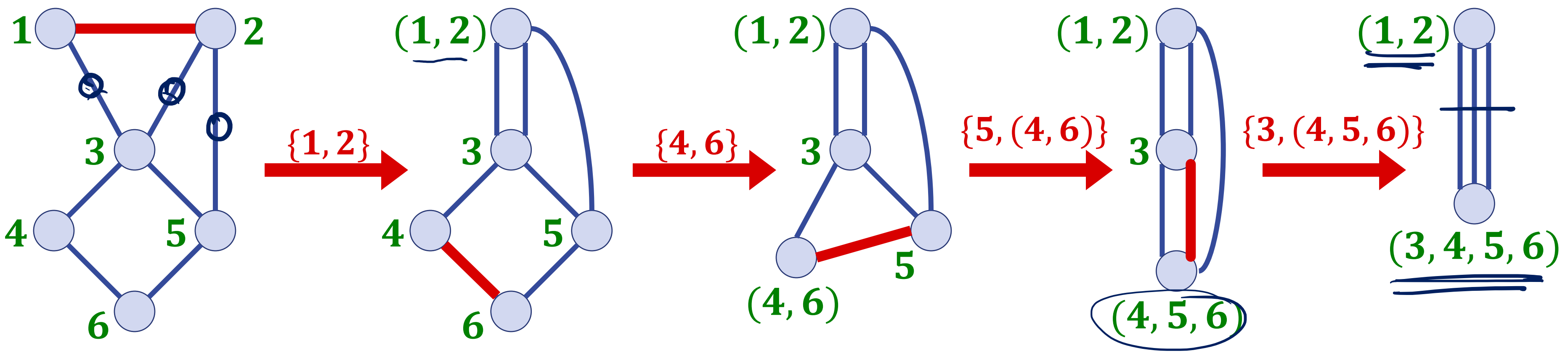
- Replace nodes u, v by new node w
- For all edges $\{u, x\}$ and $\{v, x\}$, add an edge $\{w, x\}$
- Remove self-loops created at node w



Properties of Edge Contractions

Nodes:

- After contracting $\{u, v\}$, the new node represents u and v
- After a series of contractions, each node represents a subset of the original nodes



Cuts:

- Assume in the contracted graph, w represents nodes $S_w \subset V$
- The edges of a node w in a contracted graph are in a one-to-one correspondence with the edges crossing the cut $(S_w, V \setminus S_w)$

Randomized Contraction Algorithm

Algorithm:

while there are > 2 nodes **do**

contract a uniformly random edge

return cut induced by the last two remaining nodes

(cut defined by the original node sets represented by the last 2 nodes)

Theorem: The random contraction algorithm returns a minimum cut with probability at least $1/O(n^2)$.

- We will show this next.

Theorem: The random contraction algorithm can be implemented in time $O(n^2)$.

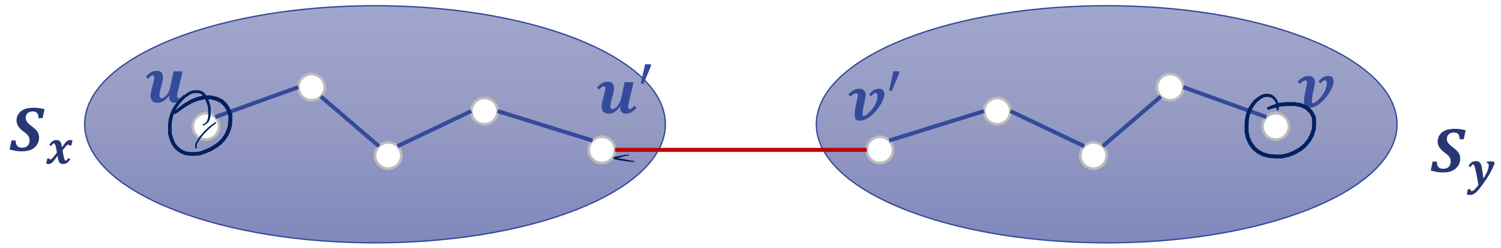
- There are $n - 2$ contractions, each can be done in time $O(n)$.
- We will see this later.

Contractions and Cuts

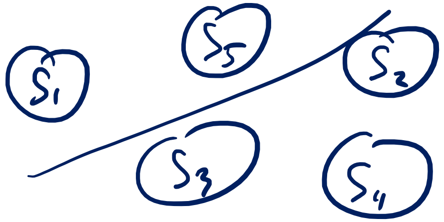
Lemma: If two original nodes $u, v \in V$ are merged into the same node of the contracted graph, there is a path connecting u and v in the original graph s.t. all edges on the path are contracted.

Proof:

- Any edge $\{x, y\}$ in the contracted graph corresponds to some edge in the original graph between two nodes u' and v' in the sets S_x and S_y represented by x and y.
- Contracting $\{x, y\}$ merges the node sets S_x and S_y represented by x and y and does not change any of the other node sets.
- The claim then follows by induction on the number of edge contractions.



Contractions and Cuts



Lemma: During the contraction algorithm, the edge connectivity (i.e., the size of the min. cut) cannot get smaller.

Proof:

- All cuts in a (partially) contracted graph correspond to cuts of the same size in the original graph G as follows:
  - For a node u of the contracted graph, let S_u be the set of original nodes that have been merged into u (the nodes that u represents)
  - Consider a cut (A, B) of the contracted graph
  - (A', B') with

$$A' \coloneqq \bigcup_{u \in A} S_u, \qquad B' \coloneqq \bigcup_{v \in B} S_v$$

    is a cut of G.
  - The edges crossing cut (A, B) are in one-to-one correspondence with the edges crossing cut (A', B').

Contraction and Cuts

Lemma: The contraction algorithm outputs a cut (A, B) of the input graph G if and only if it never contracts an edge crossing (A, B).

Proof:

1. If an edge crossing (A, B) is contracted, a pair of nodes $u \in A$, $v \in B$ is merged into the same node and the algorithm outputs a cut different from (A, B).

2. If no edge of (A, B) is contracted, no two nodes $u \in A$, $v \in B$ end up in the same contracted node because every path connecting u and v in G contains some edge crossing (A, B)

   In the end there are only 2 sets → output is (A, B)

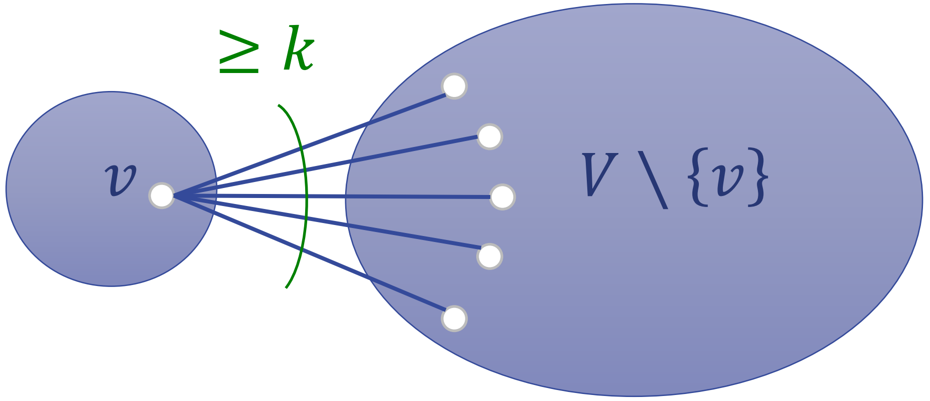
Getting The Min Cut

Theorem: The probability that the algorithm outputs a specific minimum cut is at least $2/(n(n-1)) = 1/\binom{n}{2}$.

To prove the theorem, we need the following claim:

Claim: If the minimum cut size of a multigraph G (no self-loops) is k, G has at least $kn/2$ edges.

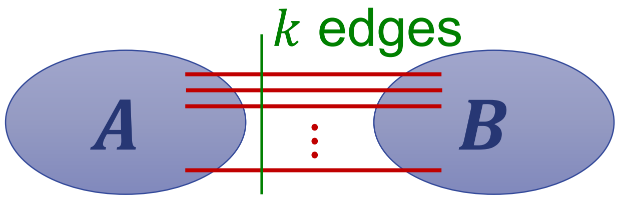
Proof:



- Min cut has size $k \Longrightarrow$ all nodes have degree $\geq k$
  - A node v of degree $< k$ gives a cut $(\{v\}, V \setminus \{v\})$ of size $< k$
- Number of edges $m = {}^1\!/_2 \cdot \sum_v \deg(v) \geq {}^1\!/_2 \cdot nk$

Getting The Min Cut

Theorem: The probability that the algorithm outputs a specific minimum cut is at least $2/n(n-1)$.



Proof:

- Consider a fixed min cut (A, B), assume (A, B) has size k
- The algorithm outputs (A, B) iff none of the k edges crossing (A, B) gets contracted.
- Before contraction i, there are $n + 1 - i$ nodes
  → and thus $\geq (n + 1 - i)k/2$ edges
- If no edge crossing (A, B) is contracted before, the probability to contract an edge crossing (A, B) in step i is at most

$$\frac{k}{\frac{(n+1-i)k}{2}} = \frac{2}{n+1-i}.$$

Getting The Min Cut

Theorem: The probability that the algorithm outputs a specific minimum cut is at least $2/n(n-1)$.

Proof:

- If no edge crossing (A,B) is contracted before, the probability to contract an edge crossing (A,B) in step i is at most ${}^{2}/_{n+1-i}$.

- Event $\mathcal{E}_i$: edge contracted in step i is **not** crossing (A,B)
  - Goal: show that $\mathbb{P}(\mathcal{E}_1 \cap \dots \cap \mathcal{E}_{n-2}) \geq 2/(n(n-1))$

$$\begin{aligned}
&\mathbb{P}\big(\text{alg. returns } (A,B)\big) \\
&\quad = \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2 \cap \dots \cap \mathcal{E}_{n-2}) \\
&\quad = \mathbb{P}(\mathcal{E}_1) \cdot \mathbb{P}(\mathcal{E}_2 \mid \mathcal{E}_1) \cdot \mathbb{P}(\mathcal{E}_3 \mid \mathcal{E}_1 \cap \mathcal{E}_2) \cdot \dots \cdot \mathbb{P}(\mathcal{E}_{n-2} \mid \mathcal{E}_1 \cap \mathcal{E}_2 \cap \dots \cap \mathcal{E}_{n-3})
\end{aligned}$$

$$\boxed{\mathbb{P}(\mathcal{E}_i \mid \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{i-1}) \geq 1 - \frac{2}{n+1-i} = \frac{n-i-1}{n-i+1}}$$

$$\mathbb{P}(\overline{\mathcal{E}_i} \mid \mathcal{E}_1 \cap \dots \mathcal{E}_{i-1}) \leq \frac{2}{n+1-i}$$

Getting The Min Cut

Theorem: The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$.

Proof:

- $\mathbb{P}(\mathcal{E}_i \mid \mathcal{E}_1 \cap \cdots \cap \mathcal{E}_{i-1}) \geq 1 - \frac{2}{n-i+1} = \frac{n-i-1}{n-i+1}$
- No edge crossing (A, B) contracted: event $\mathcal{E} = \bigcap_{i=1}^{n-2} \mathcal{E}_i$

$$\begin{aligned}\mathbb{P}(\mathcal{E}) &= \mathbb{P}(\mathcal{E}_1 \cap \cdots \cap \mathcal{E}_{n-2}) \\ &= \mathbb{P}(\mathcal{E}_1) \cdot \mathbb{P}(\mathcal{E}_2 \mid \mathcal{E}_1) \cdot \cdots \cdot \mathbb{P}(\mathcal{E}_{n-2} \mid \mathcal{E}_1 \cap \cdots \cap \mathcal{E}_{n-3})\end{aligned}$$

Randomized Min Cut Algorithm

Theorem: If the contraction algorithm is repeated $O(n^2 \log n)$ times, one of the $O(n^2 \log n)$ instances returns a min. cut w.h.p.

Proof:

- Probability to not get a minimum cut in $c \cdot \binom{n}{2} \cdot \ln n$ iterations:

$$\left(1 - \frac{1}{\binom{n}{2}}\right)^{c \cdot \binom{n}{2} \cdot \ln n} \leq e^{-c \ln n} = \frac{1}{n^c}$$

$\forall x \in \mathbb{R} : (1 + x) \leq e^x$

Corollary: The contraction algorithm allows to compute a minimum cut in $O(n^4 \log n)$ time w.h.p.

- It remains to show that each instance can be implemented in $O(n^2)$ time.