TDA251/DIT280: Algorithms Advanced Course

Lecture 7: Period 2, 2016

Chalmers University of Technology

Some slides adapted from Kevin Wayne's slides.

Randomized Algorithms

Pic ref: https://www.vitamindcouncil.org/wp-content/uploads/2013/11/randomly-picked-e1384190856897-620x413.jpg

Randomized Algorithms

"Algorithms that make random decisions during their execution."

- Can be faster and simpler than a deterministic (no random bits) algorithm.
- Example: worst case running time of Quicksort: $O(n^2)$.
  - with a random pivot, a running time of: $O(n \log n)$.
- More examples:
  - Symmetry breaking protocols, hashing, load balancing, cryptography.
- Typical uses include: In a standard algorithm, to reduce
  - either the running time (time complexity)
  - or the memory used (space complexity).

Randomized Algorithms

"Algorithms that make random decisions during their execution."

- Many NP-hard problems may be easy to solve for "typical" inputs.
  - One approach is to use heuristics to deal with pathological inputs.

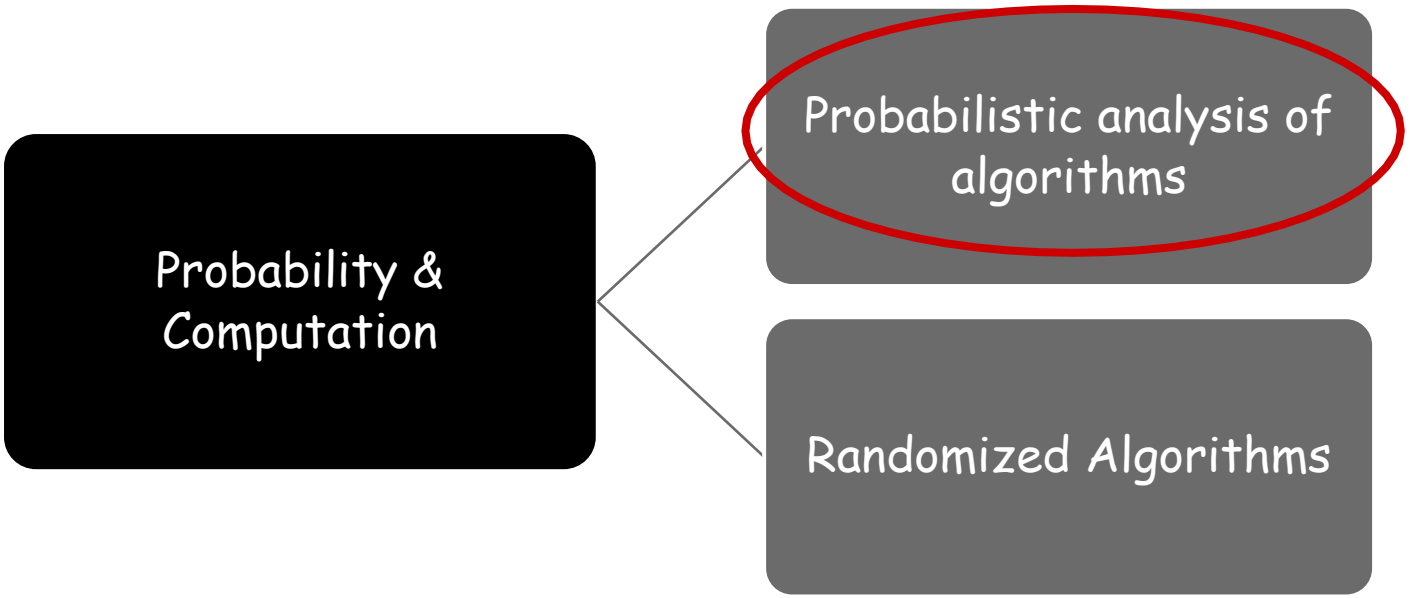
the input that is so badly arranged that it makes the algorithm to behave the worst.

  - Another approach is to use randomization (of inputs, or of algorithm) to reduce the chances of worst-case behavior.

Our focus

    - ✓ We'll focus on randomness in the algorithms.
    - No probabilistic assumptions are made on the input distributions!

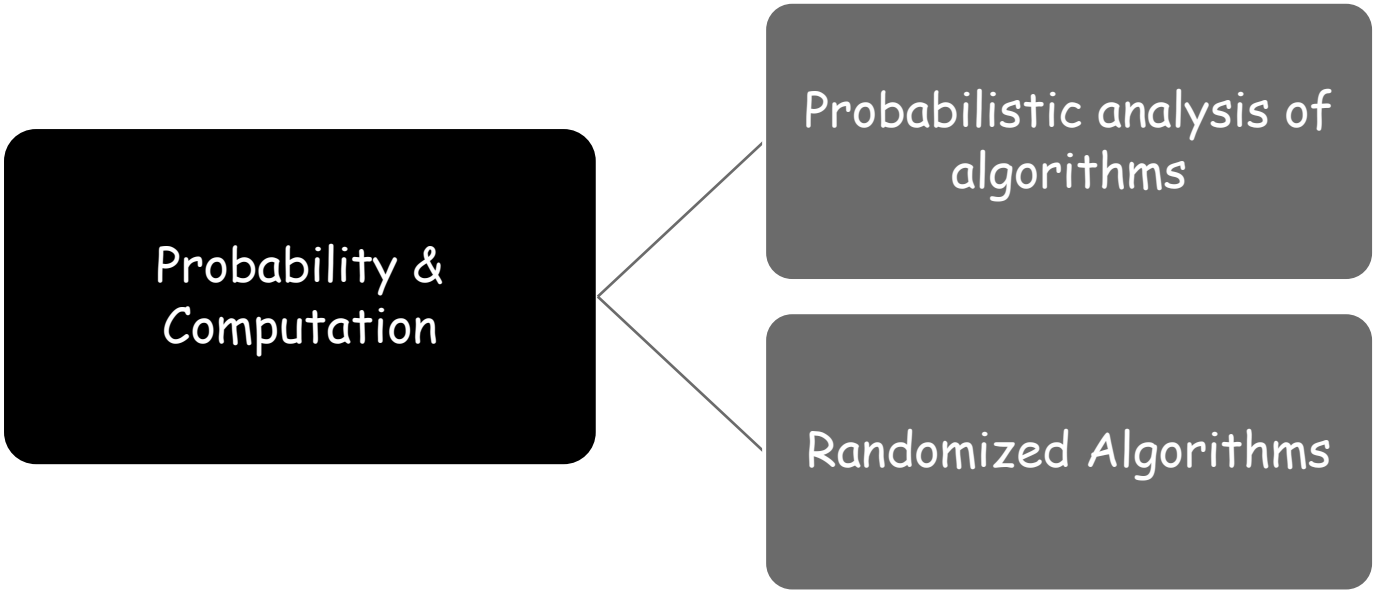
Probabilistic analysis of algorithms



Probabilistic analysis of algorithms

- Prob. theory to analyze the behavior of randomized or deterministic algorithms.
  - Example: determining the probability of a collision of a hash function.

Probabilistic analysis of algorithms



- if an algorithm makes random decisions, what about its performance?
  - ➢ it is not deterministic!
- also, a deterministic algorithm's behavior may vary with inputs.

✓ Probabilistic analysis also lets us estimate bounds on behavior.

It requires knowledge of probability theory

  - ➢ Let's start with a quick probability refresher

Some Basics of Probability Theory

Probability theory is the branch of mathematics concerned with probability, the analysis of random phenomena.

Wikipedia

Pic ref: https://stonybrook.digication.com/files/M0df463628e0f4018d8f322e4995ba854.jpg

Finite Probability Space

- If a fair coin is flipped,
- what is the probability of 'heads' ?
- It is 1/2.

- If a fair die is rolled,
- what is the probability of a '3'?
- It is 1/6.

➢ We have institutive understanding of such simple probabilities!

Most algorithmic settings (although somewhat larger and more complex)

can also be studied and easily understood if:

- We have a mathematical framework to precisely describe such statements.

Finite Probability Space

Notion of probability are usually described as follows:

- We specify a probability space which "lives" on a set Ω called sample space.
  - Ω consists of the possible outcomes of the process under consideration.
  - e.g.,
    - Single throw of a dice: $\Omega = \{1, 2, 3, 4, 5, 6\}$
    - Tossing a coin: $\Omega = \{H, T\}$
    - Tossing two coins: $\Omega = \{HH, HT, TH, TT\}$

➢ discrete Ω is the most relevant case in algorithmic contexts.

- Finite
  - As in the above examples.
- Infinite
  - Tossing a coin until we get a tail
    - $\Omega = \{T, HT, HHT, HHHT, ..., \{HHH...\}\}$.

Finite Probability Space

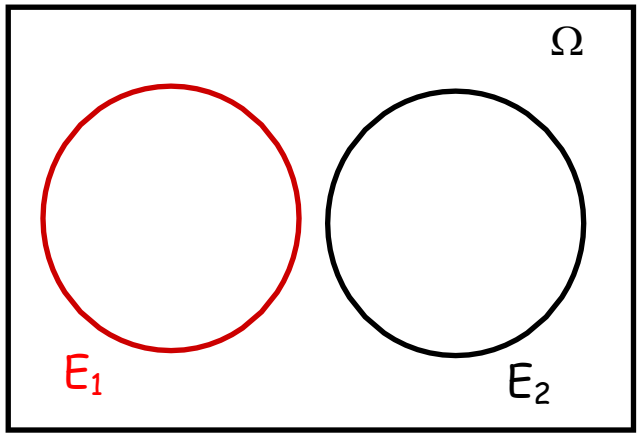
Notion of probability are usually described as follows:

- We specify a probability space which "lives" on a set Ω called sample space.
  - Ω consists of the possible outcomes of the process under consideration.
- Each element $i \in \Omega$, has nonnegative probability mass $p(i) \geq 0$, and $\sum_{i \in \Omega} p(i) = 1$.
- Subsets of Ω are called events
  - $Pr(E) = \sum_{i \in E} p(i)$
    - Consider single throw of a dice: $\Omega = \{1, 2, 3, 4, 5, 6\}$
    - A = {the number is even}; $Pr(A) = 1/6 + 1/6 + 1/6 = 3/6 = 1/2$
    - B = {the number is divisible by 3}; $Pr(B) = 2/6 = 1/3$
    - $C = \{1\}$; $Pr(C) = 1/6$ (By def. every single outcome is also an event)
- Ω and $\varnothing$ are also possible subsets, we have: $Pr(\Omega) = 1$, and $Pr(\varnothing) = 0$
- Special case: if all $i \in \Omega$ have the same probability mass: $Pr(E) = \frac{|E|}{|\Omega|}$

Finite Probability Space

Combinations (union & intersection) of events are also events

- Union (Disjunction)
  - $E = E_1$ OR $E_2 = E_1 \cup E_2$ (OR $\Leftrightarrow \cup$)
- Intersection (Conjunction)
  - $E = E_1$ AND $E_2 = E_1 \cap E_2$ (AND $\Leftrightarrow \cap$)



Mutually exclusive (disjoint) events

- $E_1 \cap E_2 = \varnothing$
- $Pr(E_1 \cup E_2) = Pr(E_1) + Pr(E_2)$
- $Pr(E_1 \cup E_2 \cup \ldots \cup E_k) = Pr(E_1) + Pr(E_2) + \ldots \cup Pr(E_k)$

Partition of the sample space

- Set of mutually exclusive events which cover the entire Ω
- $Pr(E_1 \cup E_2 \cup \ldots \cup E_k) = Pr(E_1) + Pr(E_2) + \ldots \cup Pr(E_k) = 1$

Finite Probability Space

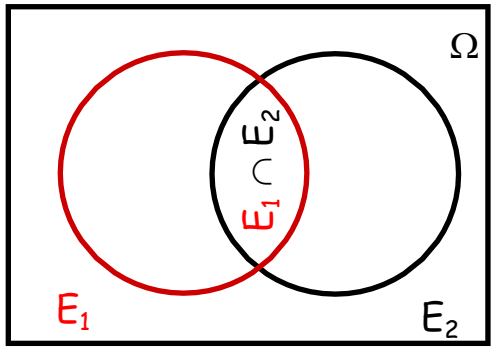
Complementary events

- E & F are complements of each other
  - if $E \cap F = \varnothing$ and $E \cup F = \Omega$.
- For an event E, define the complementary event: $\bar{E} = \Omega \backslash E$
  - $Pr(\bar{E}) = 1 - Pr(E)$.
- Complementary events are mutually exclusive.

Intersecting events

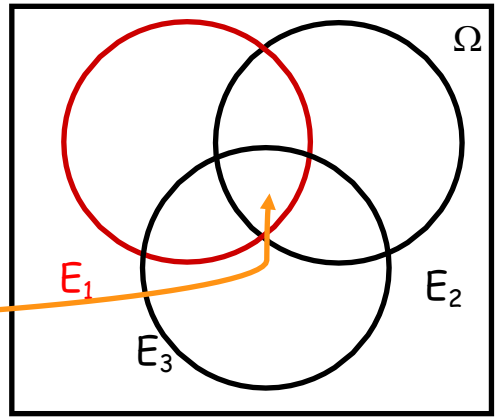
- For any events E_1, E_2, if $E_1 \cap E_2 \neq \varnothing$

  $Pr(E_1 \cup E_2) = Pr(E_1) + Pr(E_2) - Pr(E_1 \cap E_2)$

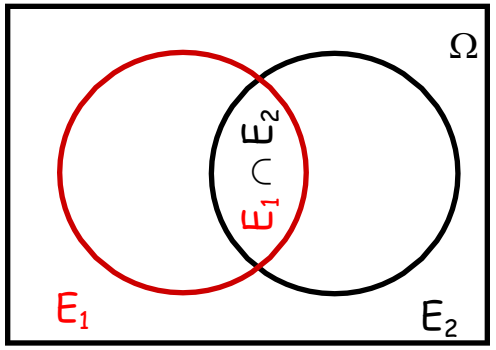


- If there are 3 events with non-empty intersections:

$$Pr(E_1 \cup E_2 \cup E_3) = Pr(E_1) + Pr(E_2) + Pr(E_3)$$
$$- Pr(E_1 \cap E_2) - Pr(E_1 \cap E_3) - Pr(E_2 \cap E_3)$$
$$+ Pr(E_1 \cap E_2 \cap E_3)$$



Finite Probability Space



Union Bound

- For any events E_1, E_2
  - $\Pr(E_1 \cup E_2) \leq \Pr(E_1) + \Pr(E_2)$

"probability that at least one of E_1, E_2 happens is always lesser than (at most equal to) the sum of their individual probabilities."

- If there are 3 events
  - $\Pr(E_1 \cup E_2 \cup E_3) \leq \Pr(E_1) + \Pr(E_2) + \Pr(E_3)$
- General form (for any finite and countable set of events)
  - $\Pr(E_1 \cup E_2 \cup \ldots \cup E_k) \leq \Pr(E_1) + \Pr(E_2) + \ldots \cup \Pr(E_k)$

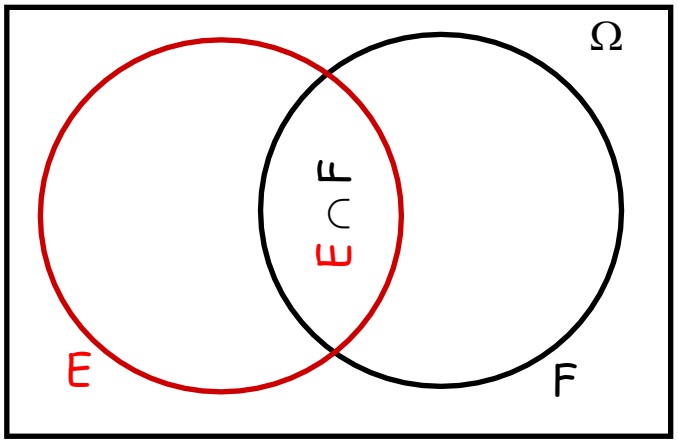
- Powerful tool for analysis of randomized algorithms
  - Used to bound the probability of a complicated event which consists of simpler events whose probabilities are easily computable.

Finite Probability Space

Conditional Probability

- We know that some event F has occurred.
- What is probability of E under this additional information?

"Of the portion of the sample space that consists of F (the event we "know" to have occurred), what fraction is occupied by E ?"



- $\Pr(E \mid F) = \dfrac{|E \cap F|}{|F|}$

$$= \frac{|E \cap F|/|\Omega|}{|F|/|\Omega|}$$

$$= \frac{\Pr(E \cap F)}{\Pr(F)}$$

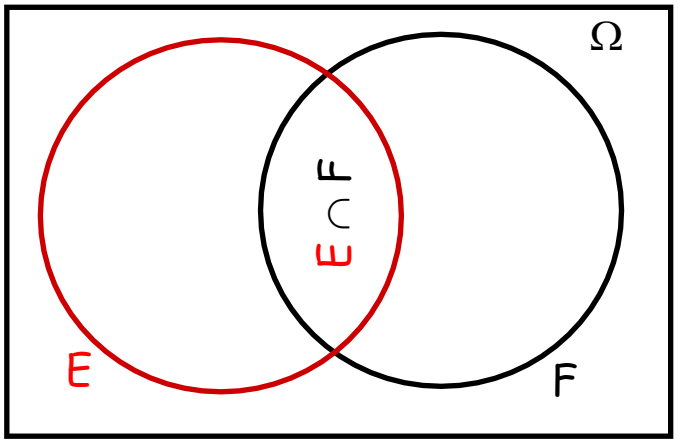
- $\Pr(F) \neq 0$

Finite Probability Space

Conditional Probability

- We know that some event F has occurred.
- What is probability of E under this additional information?

"Of the portion of the sample space that consists of F (the event we "know" to have occurred), what fraction is occupied by E ?"



- $\Pr(E \mid F) = \frac{|E \cap F|}{|F|} = \frac{|E \cap F|/|\Omega|}{|F|/|\Omega|} = \frac{\Pr(E \cap F)}{\Pr(F)}$
  - $\Pr(F) \neq 0$

A typical scenario

- Let E be a complicated event, and we want to find out Pr(E)
- Suppose $E_1, E_2, \ldots, E_k$ have positive probabilities and they partition Ω.
- Furthermore, we know $\Pr(E_i)$ and $\Pr(E \mid E_i)$ $\forall$ $i = 1, \ldots, k$.

- $\Pr(E) = \sum_{j=1 \text{ to } k} \Pr(E \mid E_i) \cdot \Pr(E_i) = \sum_{j=1 \text{ to } k} \frac{\Pr(E \cap E_i)}{\Pr(E_i)} \cdot \Pr(E_i) = \sum_{j=1 \text{ to } k} \Pr(E \cap E_i)$

Finite Probability Space

Conditional Probability & Independence of Events

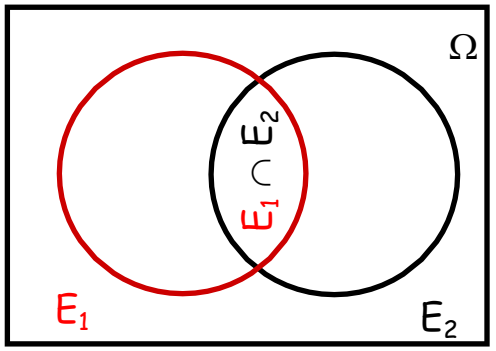
- Two events are independent if occurrence of one does not affect the likelihood of the occurrence of the other.
  - Precisely: $\Pr(E \mid F) = \Pr(E)$
- Substituting $\Pr(E \mid F) = \frac{\Pr(E \cap F)}{\Pr(F)}$ we get: $\Pr(E \cap F) = \Pr(E) \cdot \Pr(F)$
  - Independence relation is symmetric
    - Both E and F are independent of each other

Be careful!

- Independence is not always intuitive!
- For: $\Pr(E_1 \cap E_2 \cap \ldots \cap E_k) = \Pr(E_1) \cdot \Pr(E_2) \cdot \ldots \cdot \Pr(E_k)$, to hold
  - it is not enough that all distinct pairs (E_i, E_j) are independent events!
  - it requires that all possible subsets of these events are independent!
- Finally, don't confuse independent events (do not impact each other) with those of disjoint events (can not occur at the same time).

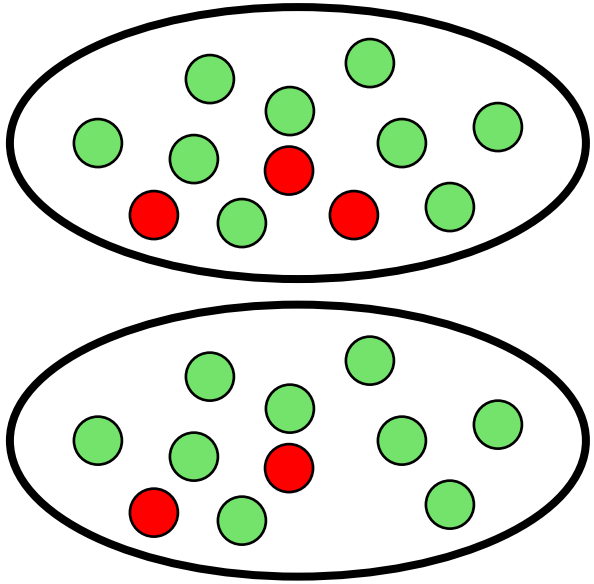
Finite Probability Space

Intersecting events & Multiplication rule



- For any events E_1, E_2, if $E_1 \cap E_2 \neq \varnothing$
  - $\Pr(E_1 \cup E_2) = \Pr(E_1) + \Pr(E_2) - \Pr(E_1 \cap E_2)$
  - $\Pr(E_1 \cap E_2)$?
- If the events are independent:
  - $\Pr(E_1 \cup E_2) = \Pr(E_1) + \Pr(E_2) - \Pr(E_1) \cdot \Pr(E_2)$
- What if the events are NOT independent?
  - $\Pr(E_1 \cap E_2) = \Pr(E_1) \cdot \Pr(E_2 \mid E_1)$
  - E_1 = first ball drawn is red
  - E_2 = second ball drawn is red
  - $\Pr(E_1 \cap E_2) = \Pr(E_1) \cdot \Pr(E_2 \mid E_1)$

    $= (3/12) \cdot (?) = (3/12) \cdot (2/11)$
  - $\Pr(E_1 \cap E_2 \cap E_3)$?
  - $\Pr(E_1 \cap E_2 \cap E_3) = \Pr(E_1) \cdot \Pr(E_2 \mid E_1) \cdot \Pr(E_3 \mid E_1 \cap E_2)$



Global Minimum Cut

Global Minimum Cut

Given: A connected, undirected graph $G = (V, E)$.

Def. A cut (A, B) of G is a partition of V into non-empty sets A & B.

- Size (cardinality) of a cut (A, B):
  - the number of edges with one end in A and the other in B.

Global Minimum Cut: A cut of minimum cardinality.

- Here "Global" is to emphasize that there is no source and sink.
  - We have seen s-t cut in network flows.

Applications. Partitioning items in a database, identify clusters of related documents, network reliability, network design, circuit design, TSP solvers.

False intuition. Global min-cut is harder than min s-t cut.

Global Minimum Cut

Brute Force:

- Try them all — exponential since there are $2^{|E|}$ subsets of edges.

Observation: the global min-cut is the minimum over all possible s-t cuts.

Adapting Ford-Fulkerson:

- Replace every edge (u, v) with two antiparallel edges (u, v) and (v, u).
  - Every directed edge has capacity 1
- Compute the min s-t cut for all pairs $s, t \in V$.

Global min-cut: min (Compute the min s-t cut for all pairs $s, t \in V$)

☛ This algorithm requires $O(n^2)$ calls to a min s-t cut (max s-t flow) solver.

Even possible with O(n) calls:

- Pick some vertex s and compute min s-v cut separating s from each other vertex $v \in V$.

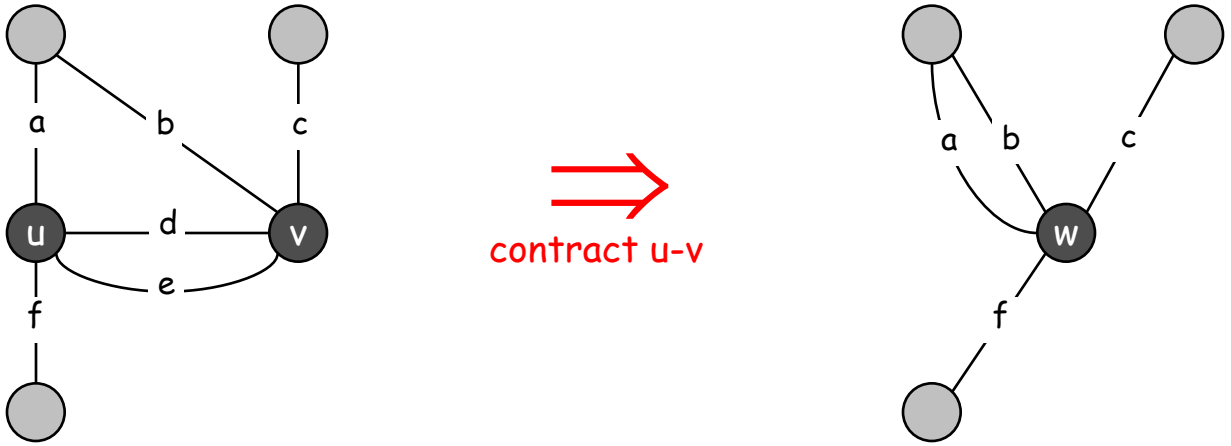
☛ any node s must appear in one of A or B.

Global Minimum Cut – Randomized Algorithm

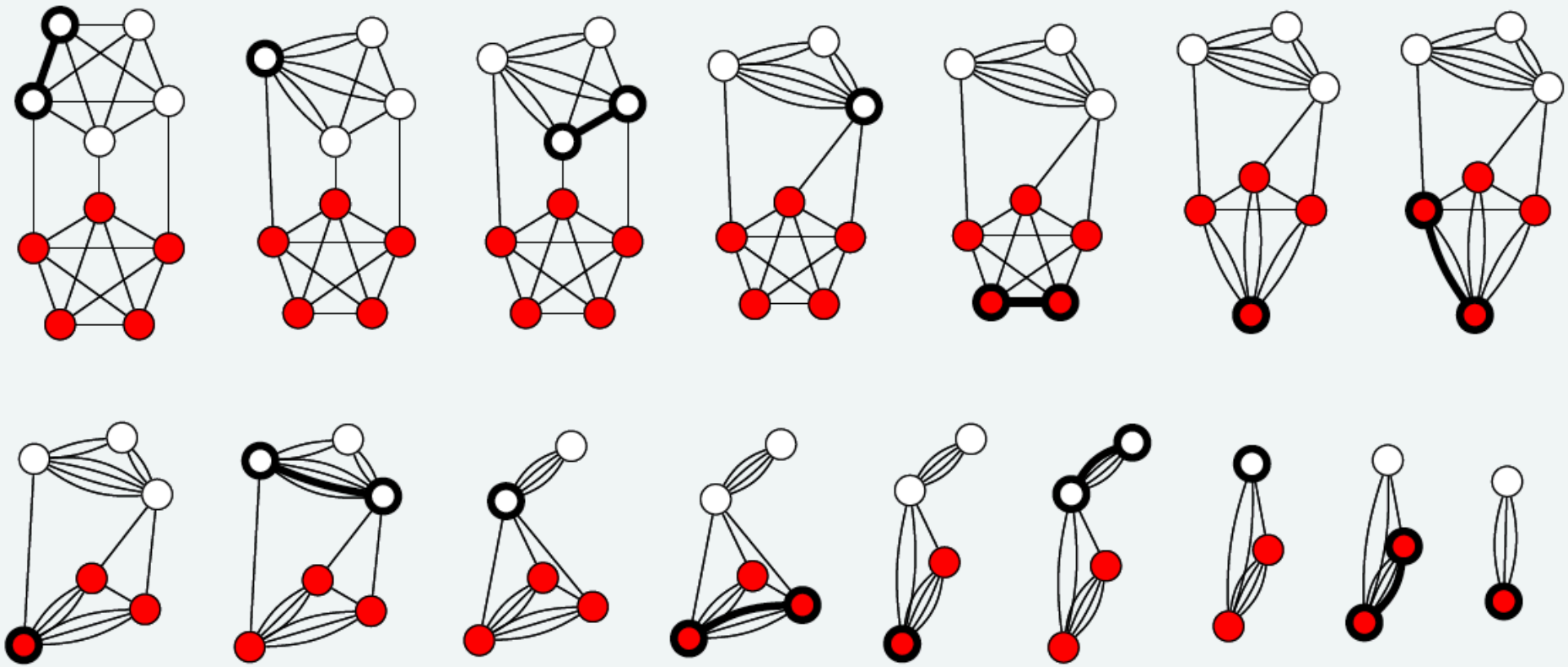
Requires access to a pseudo-random number generator

Contraction algorithm.

- Pick an edge e = (u, v) uniformly at random.
- Contract edge e.
  - replace u and v by single new super-node w
  - preserve edges, updating endpoints of u and v to w
  - keep parallel edges, but delete self-loops
- Repeat until graph has just two nodes v_1 and v_2.
- Return the cut (all nodes that were contracted to form v_1).



Global Minimum Cut – Randomized Algorithm



- The algorithm stops after n-2 iterations. (above exp. 10 nodes — 8 iteration)
- How do we know the algorithm is reliable?
- Let k = |F*| be the size of the Global Min-Cut (A*, B*).
- Precisely, we are concerned about:
  - Pr(some e ∈ F* is contracted)?
  - Pr(ALL e ∈ F* are NOT contracted)?

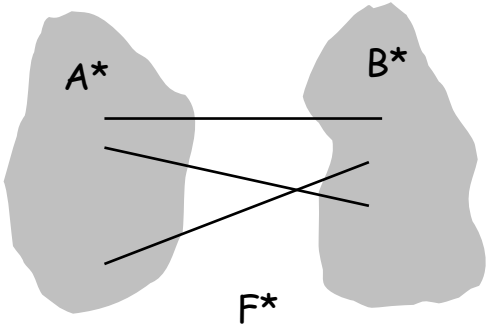
Pr(contraction algorithm finds global min-cut)

Contraction Algorithm — Success Probability

Claim. The contraction algorithm returns a global min-cut with prob $\geq 2/n^2$.

Pf. Consider a global min-cut (A^*, B^*) of G.

- Let F^* be edges with one endpoint in A^* and the other in B^*.
- Let $k = |F^*|$ = size of min cut.



- Pr(some $e \in F^*$ is contracted in FIRST iteration)

$$= \frac{\#\text{edges in the min-cut}}{\#\text{edges in } G} = \frac{k}{|E|}$$

- Every node has degree $\geq k$
  - since otherwise (A^*, B^*) would not be min-cut.

$\Rightarrow |E| \geq \frac{1}{2}kn$. ← (the $\frac{1}{2}$ is because it's two nodes to an edge)

- Pr(some $e \in F^*$ is contracted in FIRST iteration) $= \frac{k}{|E|} \leq \frac{k}{\frac{1}{2}kn}$
- Pr(some $e \in F^*$ is contracted in FIRST iteration) $\leq \frac{2}{n}$.
- Pr(some $e \in F^*$ is NOT contracted in FIRST iteration) $\geq 1 - \frac{2}{n}$.

Complementary events

Contraction Algorithm — Success Probability

Claim. The contraction algorithm returns a global min-cut with prob $\geq 2/n^2$.

Pf.

- Pr(some $e \in F^*$ is NOT contracted in FIRST iteration) $\geq 1 - \frac{2}{n}$.
- Let E_i be the event that some $e \in F^*$ is NOT contracted in i^{th} iteration.
- $\Pr(E_1) \geq 1 - \frac{2}{n}$.
- Suppose E_1 has occurred!
- What about E_2 under this additional information: $\Pr(E_2 \mid E_1) = \frac{\Pr(E_1 \cap E_2)}{\Pr(E_1)}$

Complicated

  - E_i are not independent! Why?

Contraction e = (u,v) replaces u,v with super-node w

- Think differently: E_1 has occurred implies now we have: $|E| \geq \frac{1}{2}k(n-1)$ edges.
  - Using same reasoning as in the previous slide: $\Pr(E_2 \mid E_1) \geq 1 - \frac{2}{n-1}$
- After i iterations: $\Pr(E_i \mid E_1 \cap E_2 \cap \ldots \cap E_{i-1}) \geq 1 - \frac{2}{n-i+1}$
  - For every i, the k remains unchanged while edges: $|E| \geq \frac{1}{2}k(n-i+1)$ edges.

Contraction Algorithm — Success Probability

Claim. The contraction algorithm returns a global min-cut with prob $\geq 2/n^2$.

Pf.

- $\Pr(E_1) \geq 1 - \frac{2}{n}$.
- $\Pr(E_1 \mid E_2) \geq 1 - \frac{2}{n-1}$
- After i iterations: $\Pr(E_i \mid E_1 \cap E_2 \cap \ldots \cap E_{i-1}) \geq 1 - \frac{2}{n-i+1}$

Why multiplication?

$$\Pr(\text{success}) = \Pr(E_1 \cap E_2 \cap \ldots \cap E_{n-2})$$

$$= \Pr(E_1) \cdot \Pr(E_2 \mid E_1) \cdot \Pr(E_3 \mid E_1 \cap E_2) \cdot \ldots \cdot \Pr(E_{n-2} \mid E_1 \cap E_2 \cap \ldots \cap E_{n-3})$$

$$\geq \left(1 - \frac{2}{n}\right) \cdot \left(1 - \frac{2}{n-1}\right) \cdot \left(1 - \frac{2}{n-2}\right) \cdot \ldots \cdot \left(1 - \frac{2}{4}\right) \cdot \left(1 - \frac{2}{3}\right)$$

$$= \left(\frac{n-2}{n}\right) \cdot \left(\frac{n-3}{n-1}\right) \cdot \left(\frac{n-4}{n-2}\right) \cdot \ldots \cdot \left(\frac{2}{4}\right) \cdot \left(\frac{1}{3}\right)$$

$$= 2\left(\frac{(n-2)!}{n!}\right) = \left(\frac{2 \cdot 1}{n(n-1)}\right)$$

$$\geq 2/n^2$$

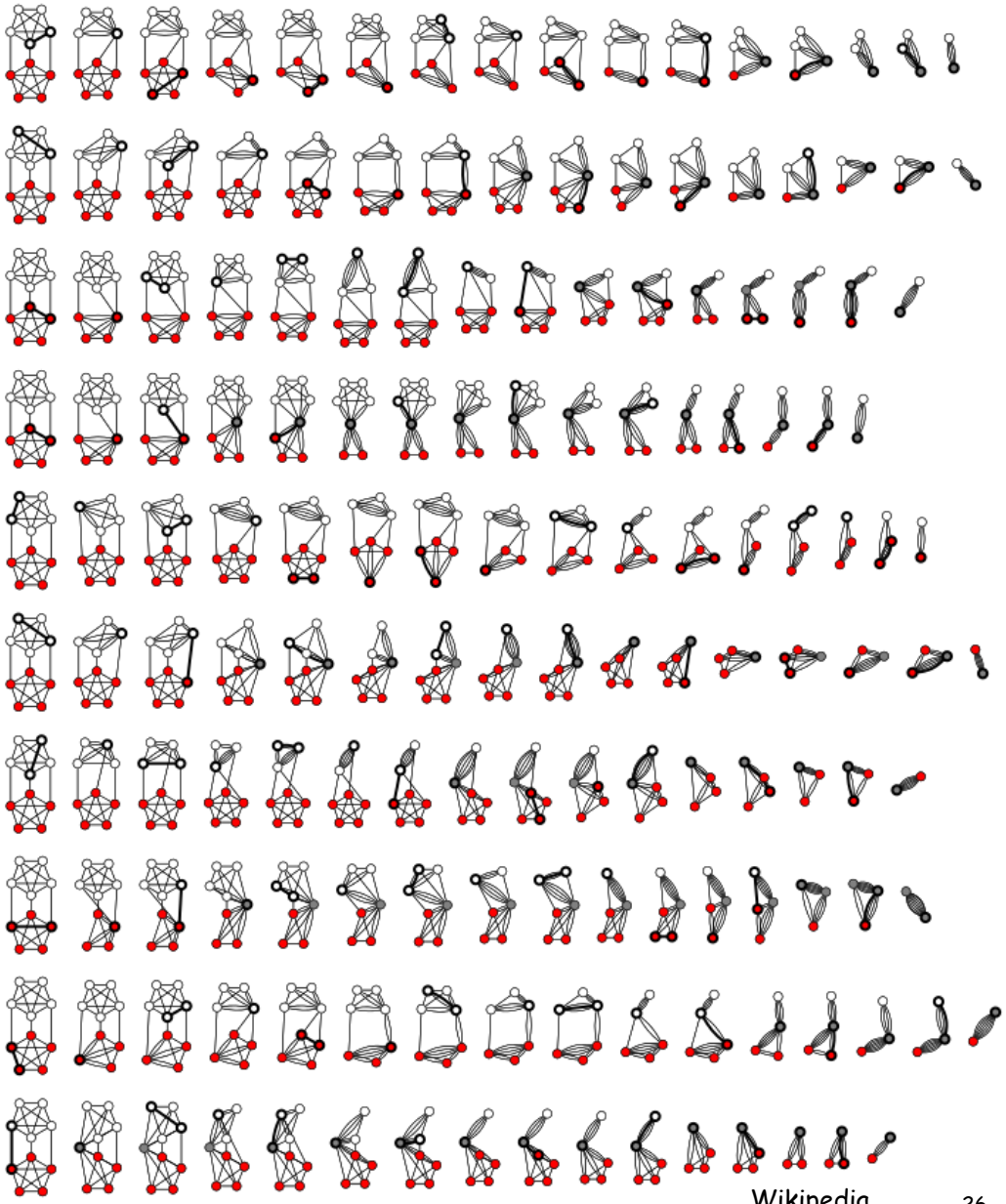
■

Contraction Algorithm — Success Probability Amplification

- Probability of success, $2/n^2$ is small!
- What happens for large n?
  - goes to zero as $n \to \infty$
- Idea:
  - Run the algorithm several times and chose the smallest min-cut.

> - 10 repetitions.
> - ✓ Found the min-cut of size 3 at the 5^{th} repetition.

- What about Pr(success)?
  - If we run the algorithm t times?
  - How does it boost Pr(success)?



Wikipedia

Contraction Algorithm — Success Probability Amplification

- What about the Pr(success)?
  - If we run the algorithm t times?
  - How does it boost the Pr(success)?
- Pr(algorithm fails the first time) $\leq \left(1 - \frac{2}{n^2}\right)$
- Pr(algorithm fails the first time and the second time) $\leq \left(1 - \frac{2}{n^2}\right)^2$
  - ✓ Each run is independent of the other!
- Pr(algorithm fails t consecutive runs) $\leq \left(1 - \frac{2}{n^2}\right)^t$
- Pr(algorithm succeeds after t times) $\geq 1 - \left(1 - \frac{2}{n^2}\right)^t$
  - How to choose a good value for t?

Contraction Algorithm — Success Probability Amplification

Amplification. To amplify the probability of success, run the O(m) contraction algorithm many times.

Claim. If we repeat the contraction algorithm $t = n^2 \ln n$ times with independent random choices, the probability of failing to find the global min-cut is at most $1/n^2$.

Pf. By independence, the probability of failure is at most

$$\left(1-\frac{2}{n^2}\right)^{n^2 \ln n} = \left[\left(1-\frac{2}{n^2}\right)^{\frac{1}{2}n^2}\right]^{2\ln n} \leq \left(e^{-1}\right)^{2\ln n} = \frac{1}{n^2}$$

$(1 - 1/x)^x \leq 1/e$

▪

➡ Prob(success) $\geq \left(1 - \frac{1}{n^2}\right)$

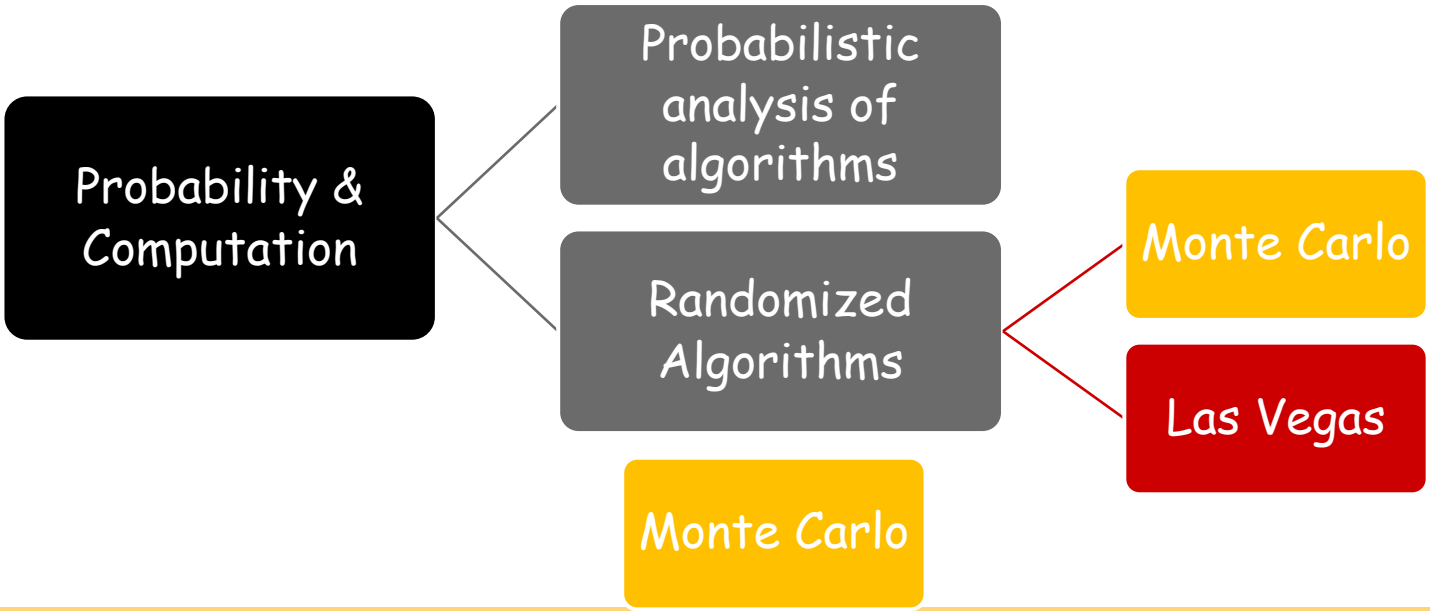
- Want to be independent of n?
  - Chosing $t = n^2 / 2$, achieves Prob(failure) $\leq 1/e$

Contraction Algorithm — Summarizing the approach

"Achieved a randomized algorithm which is simpler than a deterministic algorithm."

- Identify the bad event
  - some $e \in F^*$ is contracted
- Find its probability
  - Pr(some $e \in F^*$ is contracted)
- Bound that it DOES NOT happen
  - Pr(ALL $e \in F^*$ are NOT contracted) = Pr(success)
- Amplify the probability of success
  - run the O(m) contraction algorithm many times.

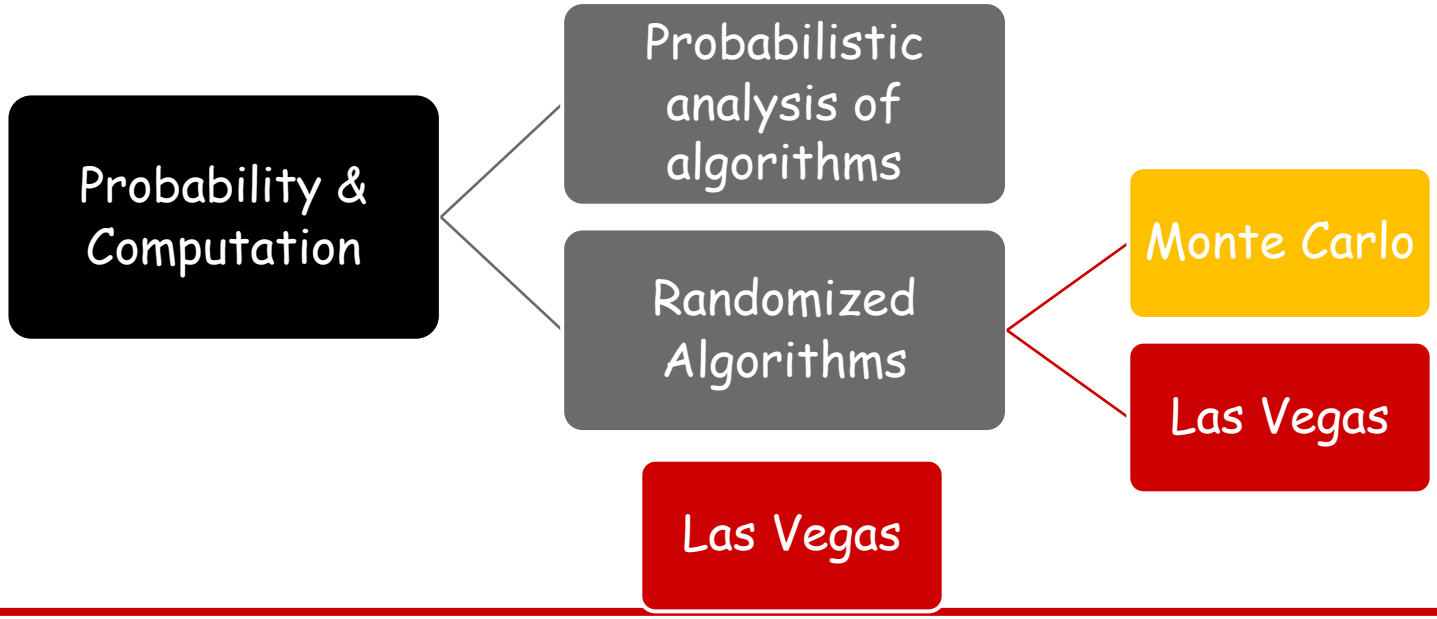
Randomized Algorithms — Two main types!



It may produce incorrect answer!

- runs for a fixed number of steps for each input
- and produces an answer that is correct with a bounded probability.
- By running it many times, we can make the failure probability arbitrarily small at the *expense of running time*.
- Randomized Global min-cut is a Monte Carlo Algorithm.

Randomized Algorithms — Two main types!



Always gives the correct solution!

- the time complexity varies on the input.
- guarantees an upper bound for the worst-case ➡ Running time is bounded.

✓ Quicksort (random pivot) is a Las Vegas algorithm.

In the language of prob. theory:

- Running time for each input is a random variable whose expectation is bounded.

Topic for next lecture: Random variables and their Expectation!