Classification of randomized algorithms

1) rand. alg. that never fail $\rightarrow$ "LAS VEGAS" alg.
e.g. randomized Quicksort

$\forall i \in I, \ A_R(i) = s$ s.t. $(i, s) \in \Pi$

$\Pi \subseteq I \times S$
decision problem

obs.: s may not be the same $\forall i$

randomness comes into play in the analysis of the complexity

$\forall n$, $T(n)$ is a <u>random variable</u>, of which we usually study $E[T(n)]$ or

$Pr(T(n) > c \cdot f(n)) \leq \frac{1}{n^k}$

$\Rightarrow T(n) = O(f(n))$ "with high probability"

space of probabilities = random choices made by the algorithm

Do not confuse this with the probabilistic analysis of deterministic algorithm, where the space of probabilities = distribution of the inputs

2) rand. alg. that may fail $\rightarrow$ "MONTE CARLO" alg.
e.g. verifying polinomial identities
$i \in I$ it's possible that $A_R(i) = s$ s.t. $(i,s) \notin \Pi$
we study $Pr((i,s) \notin \Pi)$ as a function of $|i| = n$
$\rightarrow$ family of random variables
moreover, even $T(n)$ may be a random variable
for decision problems, these algoritms can be divided into
- one-sided : they may fail only on one answer
- two-sided : they may fail in both answers

In this course we'll see 1 LAS VEGAS alg. and 1 MONTE CARLO alg.

$\hookrightarrow$ Randomized Quicksort
we'll see a high probability analysis

$\hookrightarrow$ Karger's algorithm for Minimum Cut
(again, with an analysis in high probability)

Def.: given $\Pi \subseteq I \times S$ an algorithm A_Π has complexity $T(n) = O(f(n))$ with high probability (w.h.p.) if $\exists$ constants $c, d > 0$ such that $\forall i \in I, |i| = n$,
$Pr(A_\Pi(i) \text{ terminates in } > c \cdot f(n) \text{ steps}) \leq \frac{1}{n^d}$

idea: prob. $\to 1$ as $n \to +\infty$

$\hookrightarrow O(f(n)) > 1 - \frac{1}{n^d}$

Def.: given $\Pi \subseteq I \times S$ an algorithm A_Π is correct <u>with high probability</u> (w.h.p.) if $\exists$ constant $d > 0$ s.t. $\forall i \in I$, $|i| = n$, $Pr\left((i, A_\Pi(i)) \notin \Pi\right) \leq$
$\leq \frac{1}{n^d}$

high prob. $\Longrightarrow$ expectation:

Exercise: Assume that

1) A_Π LAS VEGAS, with $T_{A_\Pi}(n) = O(f(n))$ w.h.p.; in particular, $Pr\left(T_{A_\Pi}(n) > c \cdot f(n)\right) \leq$
$\leq \frac{1}{n^d}$

2) A_Π has a worst-case deterministic complexity $O(n^a)$ $a \leq d$ $\forall n$

Show that $E\left[T_{A_\Pi}(n)\right] = O(f(n))$

apply the following:

Markov's lemma: let T be a non-negative, bounded ($= \exists\, b \in \mathbb{N}$ s.t. $\Pr(T > b) = 0$), integer random variable. Then $\forall t$ s.t. $0 \leq t \leq b$

$$t \cdot \Pr(T \geq t) \leq E[T] \leq t + (b-t)\Pr(T \geq t)$$

Karger's algorithm for Minimum Cut (1993)

cut of minimum size; in other words, it's the minimum number of edges whose removal disconnects the graph

applications: network reliability, war, ...

We'll actually solve a more general problem: minimum cut on multigraphs (i.e., multiple edges between two vertices are allowed)

Def.: multiset = collection of objects with repetitions allowed

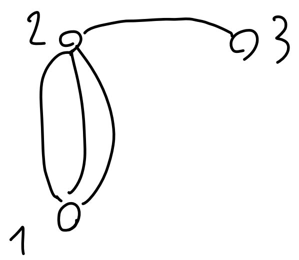
$$S = \{\{ \text{objects} \}\}$$

$\forall$ object $\sigma \in S$ $\quad m(\sigma) \in \mathbb{N} \setminus \{0\}$

↳ multiplicity: how many copies of σ are in S

Def.: a <u>multigraph</u> $\mathcal{G} = (V, \mathcal{E})$ s.t. $V \subseteq \mathbb{N}$, and $\mathcal{E}$ is a multiset of elements (u,v) $u \neq v$

example:



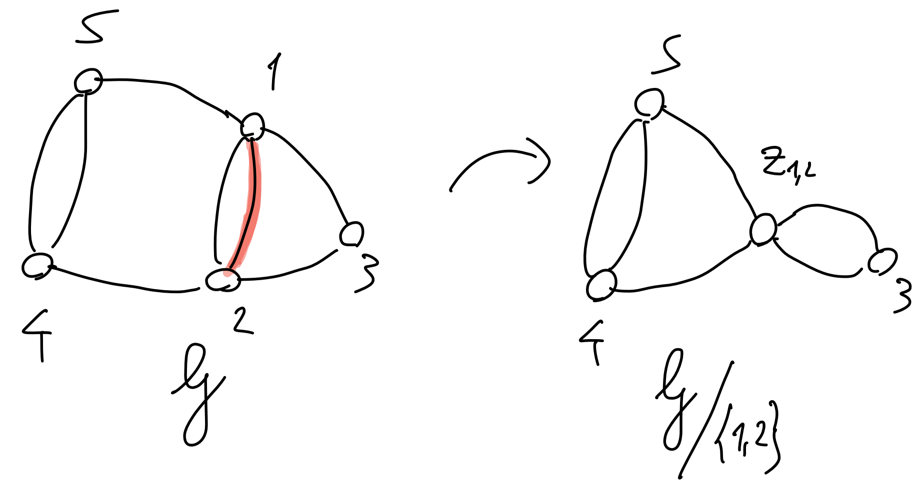
a simple graph $G = (V, E)$ is also a multigraph

Def.: given $\mathcal{G} = (V, \mathcal{E})$ connected, a <u>cut</u> $C \subseteq \mathcal{E}$ is a multiset of edges s.t. $\mathcal{G}' = (V, \mathcal{E} \setminus C)$ is not connected

Karger's algorithm

- choose an edge at random
- "contract" the 2 vertices of that edge, removing all the edges incident both vertices
- repeat until only 2 vertices remain: return the edges between them

Example :



Def.: given $\mathcal{G} = (V, \mathcal{E})$ and $e = (u,v) \in \mathcal{E}$, the <u>contraction of $\mathcal{G}$ with respect to e</u>, $\mathcal{G}/_e = (V', \mathcal{E}')$, is the multigraph with

$$V' = V \setminus \{u,v\} \cup \{z_{u,v}\} \qquad (z_{u,v} \notin V)$$

$$\mathcal{E}' = \mathcal{E} \setminus \{\{ (x,y) : (x=u) \text{ or } (x=v) \}\}$$
$$\cup \{\{ (z_{u,v}, y) : (u,y) \in \mathcal{E} \text{ or } (v,y) \in \mathcal{E},\ y \neq u \text{ and } y \neq v \}\}$$

$$|V'| = |V| - 1$$
$$|\mathcal{E}'| = |\mathcal{E}| - m(e) \leq |\mathcal{E}| - 1$$

```
FULL_CONTRACTION(G = (V, E))
  for i = 1 to n-2 do
    e ← RANDOM(E)
    G' = (V', E') ← G/e
    V ← V'
    E ← E'
  return |E|
```

```
KARGER(G, k)
  min ← +∞
  for i = 1 to k do
    t ← FULL_CONTRACTION(G)
    if t < min then
      min ← t
  return min
```

repeats FULL_CONTRACTION several (k) times to reduce the probability of error (as in verifying polinomial identities)

to be determined by the analysis