Quicksort (Hoare, 1959)

To sort $a_1, a_2, \ldots, a_n$: If $n > 1$,

1. Choose $p \sim \text{Unif}(1, n)$.
2. Let $L = \{a_i \mid a_i < a_p\}$
   $E = \{a_i \mid a_i = a_p\}$
   $G = \{a_i \mid a_i > a_p\}$.
3. Recursively sort and output L.
   Output E.
   Recursively sort and output G.

If you're unlucky (recursively), then the running time is $\sim n^2$.

What is the expected running time? $O(n \log n)$

Another probabilistic Min Cut algorithm (Karger 1993).

Let G be an undirected multigraph. (There may be multiple edges connecting vertices u and v.) A cut is a set of edges whose removal disconnects G ($\exists u, v$ with no path from u to v.). A min-cut is a cut of minimum cardinality. If G represents a communication network, then a min-cut of G is the smallest number of communication links whose failure disrupts communication. So you want the min-cut to be large.