An Important Recurrence: The Master Theorem

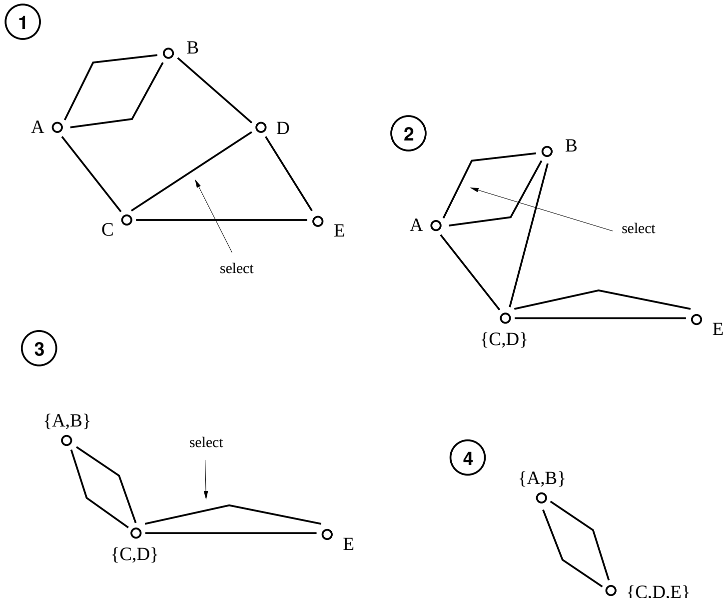
- Let $a \in \mathbb{N}, b, d, \alpha, k \in \mathbb{R}$, with $a \geq 1, b > 1, d \geq 0, \alpha > 0, k \geq 0$
- Let $g(n)$ be given by

$$g(n) \leq \begin{cases} \alpha & \text{if } n < \frac{b}{b-1}(d+1) \\ a \cdot g(\lfloor n/b + d \rfloor) + \alpha n^k & \text{otherwise} \end{cases}$$

- Let $\lambda = \log_b a$
- Then

$$g(n) = \begin{cases} O(n^k) & \text{if } \lambda < k \\ O(n^k \log n) & \text{if } \lambda = k \\ O(n^\lambda) & \text{if } \lambda > k \end{cases}$$

Illustration



A min-cut Algorithm

1. $H := G$
2. while H has more than two nodes do
3.   choose an edge (u,v) uniformly at random from the edges in H.
4.   contract (u, v) in H
5. let C be the edges of H
6. output C

A variant of the algorithm

Input: A multigraph $G = (V, E)$ with n nodes

Algorithm: FastCut(G)

1. Let n be the number of nodes of G.
2. If $n \leq 11$ then Compute the min-cut by brute-force enumeration;
3. otherwise,
4. $t := \lceil 1 + n/\sqrt{2} \rceil$
5.   perform two independent contraction sequences: obtain graphs H_1 and H_2 each with t nodes
6.   call FastCut(H_1) and FastCut(H_2), independently
7.   output the smaller of the tow returned cuts

Summary

- FastCut has running time $O(n^2 \log n)$
- FastCut has success probability of at least $\Omega(1/\log n)$

- Repeating the algorithm $\log n$ times we get a randomized algorithm with running time $O(n^2 \log^2 n)$ and constant failure probability.

Las Vegas to Monte Carlo

- Let $\alpha > 1.$
- Suppose we are given a Las Vegas algorithm A with expected running time $f(n)$ for inputs of size n.

- Consider algorithm B which runs A for $\alpha f(n)$ steps and then terminates.

- B is a Monte Carlo algorithm with failure probability at most $1/\alpha$

Las Vegas Algorithm:

- always gives the correct solution.
- running time is a random variable.
- Example: Quick-sort.

Monte Carlo Algorithm:

- may produce incorrect solutions.
- has a fixed deterministic running time.
- Example: Min-Cut Algorithm.
- For decision problems there are two kinds:

  one-sided errors: either *yes* or *no* answers always correct.

  two-sided errors: both have non-zero probability to be wrong.

Monte Carlo to Las Vegas

- Suppose we are given a Monte Carlo algorithm A with running time $f(n)$ and failure probability $p(n)$.
- Assume we can check to correctness of a solution in time $g(n)$.

- Consider algorithm B which repeatedly runs algorithm A and checks the solution until a correct solution is found.

- B is a Las Vegas algorithm with expected running time at most $(f(n) + g(n))/p(n)$.