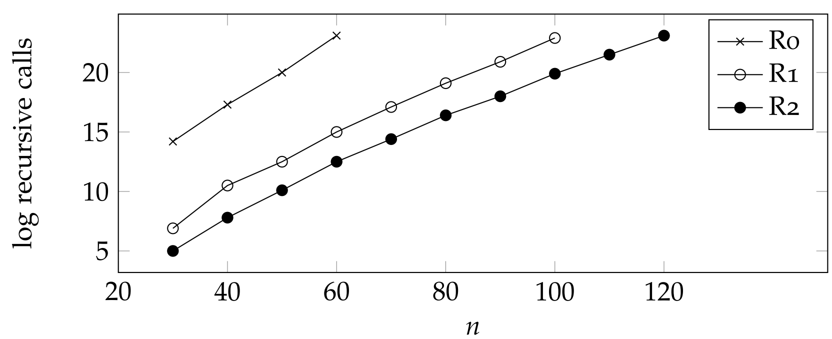
Independent Set Lab Report

by Marcus Kindberg

Correctness

For the produced set to be maximal every node must either be marked or have a marked neighbour. Algorithm R1 correctly computes $\alpha(G)$ because if a node only has one neighbour either it or its neighbour must be marked and since marking the node itself doesn't add any constraints to the rest of the graph (while marking its neighbour does) this is better.
Algorithm R2 correctly computes $\alpha(G)$ because if all three nodes are connected the case is the same as in R1 that one must be marked and it's better to mark the node itself. If the neighbours aren't connected at least one of them will get marked (for similar reasons as R1 works) and at most two will be marked (both children). Therefore one node is kept to symbolize the node that may or may not be marked.

Empirical Running time



Experiments.

The running times of algorithm R_0, R_1, and R_2 appear to be $O(1.22^n)$, $O(1.17^n)$ and $O(1.15^n)$, respectively.

Theoretical Upper Bound

Denote be $T_i(n)$ the worst runtime of algorithm Ri on *any* graph on n vertices. Note that $T_i(n)$ is a non-decreasing function of n. For R_0 we can conclude that

$$\begin{aligned} T_0(n) &\leq \max(T_0(n-1), T_0(n-1) + T_0(n-1-d_{\text{max}})) \\ &\leq T_0(n-1) + T_0(n-2) \end{aligned}$$

with d_{max} the degree of the vertex we branch on. The hard part is the one when there are no isolated vertices, in which case the vertex u we are branching on has at least one neighbor.

For R1 we have that

$$\begin{aligned} T_1(n) &\leq \max(T_1(n-1), T_1(n-2), T_1(n-1) + T_1(n-1-d_{\max})) \\ &\leq T_1(n-1) + T_1(n-3) \end{aligned}$$

For R2 we have that

$$\begin{aligned} T_2(n) &\leq \max(T_2(n-1), T_1(n-2), T_1(n-3), T_2(n-1) + T_2(n-1-d_{\max})) \\ &\leq T_2(n-1) + T_2(n-4) \end{aligned}$$

Worst Case Upper Bound The running times of algorithm R_0, R_1, and R_2 are in $O(1.62^n)$, $O(1.47^n)$, and $O(1.38^n)$, respectively.