

```

RegWidthColoringStep(top) begin
  if top >= n then
    completeColorings := completeColorings + 1;
    regWidth := totalRegisterWidth(ColorStack);
    if minRegWidth > regWidth then
      optimalColors := colorStack; minRegWidth := regWidth;
      if not MeetOptimizationTimeConstraint() then exit;
    end if
    return;
  end if
  for c in minColor(top) to maxColor(top) do
    colorStack(top) := c;
    if RegWidthLowerBound(colorStack, asap, alap) >= minRegWidth then
      cutBranches := cutBranches + 1; continue;
    end if
    if top < n-1 then
      oper := order(top+1);
      minC := estimateMinConflictColor(ColorStack, top, oper, ConflictRelation);
      maxC := estimateMaxConflictColor(ColorStack, top, oper, ConflictRelation);
      minP := estimateMinNonConflictColor(ColorStack, top, oper,
        NonConflictRelation);
      maxP := estimateMaxNonConflictColor(ColorStack, top, oper,
        NonConflictRelation);
      minColor(top+1) := maximum(asap(order(top+1)), minC+1, minP);
      maxColor(top+1) := minimum(alap(order(top+1)), maxC-1, maxP);
      if minColor(top+1) > maxColor(top+1) then continue; end if
    end if
    coloringStep(top+1);
  end for
end

```