On-Line Ramsey Theory in Bounded Degree Graphs

Douglas B. West

Department of Mathematics
University of Illinois at Urbana-Champaign
west@math.uiuc.edu
http://www.math.uiuc.edu/~west/pubs/publink.html

Joint work with
Jane Butterfield, Tracy Grauman, Bill Kinnersley, Kevin Milans, Christopher Stocker
The Problem

Graph Ramsey theory = a game
**Builder** presents a graph; **Painter** 2-colors the edges.
**Builder** wins if a monochromatic $G$ is produced.
The Problem

Graph Ramsey theory = a game
Builder presents a graph; Painter 2-colors the edges. 
Builder wins if a monochromatic $G$ is produced.

“Arrow” $H \rightarrow G \iff$ Builder wins by playing $H$. 
The Problem

Graph Ramsey theory = a game
Builder presents a graph; Painter 2-colors the edges.
Builder wins if a monochromatic $G$ is produced.

“Arrow” $H \rightarrow G \iff$ Builder wins by playing $H$.

On-line Graph Ramsey theory = a stronger Builder
Builder presents one edge at a time; Painter colors it.
Builder wins if a monochromatic $G$ is produced.
The Problem

Graph Ramsey theory = a game
Builder presents a graph; Painter 2-colors the edges. 
Builder wins if a monochromatic $G$ is produced.

“Arrow” $H \rightarrow G \Leftrightarrow$ Builder wins by playing $H$.

On-line Graph Ramsey theory = a stronger Builder
Builder presents one edge at a time; Painter colors it.
Builder wins if a monochromatic $G$ is produced.

Idea: Restrict Builder to play on a family $\mathcal{H}$.
After every move, the graph presented so far lies in $\mathcal{H}$. 
The Problem

Graph Ramsey theory = a game
Builder presents a graph; Painter 2-colors the edges.
Builder wins if a monochromatic $G$ is produced.

“Arrow” $H \rightarrow G \iff$ Builder wins by playing $H$.

On-line Graph Ramsey theory = a stronger Builder
Builder presents one edge at a time; Painter colors it.
Builder wins if a monochromatic $G$ is produced.

Idea: Restrict Builder to play on a family $\mathcal{H}$.
After every move, the graph presented so far lies in $\mathcal{H}$.
This defines the on-line Ramsey game $(G, \mathcal{H})$.
Can Builder playing on $\mathcal{H}$ force a monochromatic $G$?
Prior Work

Grytczuk–Hałuszcak–Kierstead [2004] (ElJC)
Prior Work

Grytczuk–Hałuszczak–Kierstead [2004] (ElJC)

\( \chi(G) \leq k, \mathcal{H} = k\)-colorable graphs \( \Rightarrow \) Builder wins.

\( G \) a forest, \( \mathcal{H} = \{\text{forests}\} \) \( \Rightarrow \) Builder wins.
Prior Work

Grytczuk–Hałuszczak–Kierstead [2004] (ElJC)

\( \chi(G) \leq k, \mathcal{H} = k\)-colorable graphs \( \Rightarrow \) Builder wins.

\( G \) a forest, \( \mathcal{H} = \{\text{forests}\} \Rightarrow \) Builder wins.

\( G = C_3, \mathcal{H} = \{\text{outerplanar}\} \Rightarrow \) Painter wins.

\( G = C_3, \mathcal{H} = \{\text{planar 2-degenerate}\} \Rightarrow \) Builder wins.
Prior Work

Grytczuk–Hałuszczak–Kierstead [2004] (ElJC)

$\chi(G) \leq k$, $\mathcal{H} = k$-colorable graphs $\Rightarrow$ Builder wins.

$G$ a forest, $\mathcal{H} = \{\text{forests}\}$ $\Rightarrow$ Builder wins.

$G = C_3$, $\mathcal{H} = \{\text{outerplanar}\}$ $\Rightarrow$ Painter wins.

$G = C_3$, $\mathcal{H} = \{\text{planar 2-degenerate}\}$ $\Rightarrow$ Builder wins.

$G = C_n$, $\mathcal{H} = \{\text{planar}\}$ $\Rightarrow$ Builder wins.

$G = P_n \lor K_1$, $\mathcal{H} = \{\text{planar}\}$ $\Rightarrow$ Builder wins. (extra idea)
Prior Work

Grytczuk–Hałuszczak–Kierstead [2004] (ElJC)

\( \chi(G) \leq k, \mathcal{H} = k \)-colorable graphs \( \Rightarrow \) Builder wins.

\( G \) a forest, \( \mathcal{H} = \{ \text{forests} \} \) \( \Rightarrow \) Builder wins.

\( G = C_3, \mathcal{H} = \{ \text{outerplanar} \} \) \( \Rightarrow \) Painter wins.

\( G = C_3, \mathcal{H} = \{ \text{planar 2-degenerate} \} \) \( \Rightarrow \) Builder wins.

\( G = C_n, \mathcal{H} = \{ \text{planar} \} \) \( \Rightarrow \) Builder wins.

\( G = P_n \lor K_1, \mathcal{H} = \{ \text{planar} \} \Rightarrow \) Builder wins. (extra idea)

**Conj.** On planar graphs, **Builder** wins if and only if \( G \) is outerplanar.
Prior Work

Grytczuk–Hałuszcak–Kierstead [2004] (ElJC)

\( \chi(G) \leq k, \mathcal{H} = k\)-colorable graphs \( \Rightarrow \) Builder wins.

G a forest, \( \mathcal{H} = \{\text{forests}\} \Rightarrow \) Builder wins.

\( G = C_3, \mathcal{H} = \{\text{outerplanar}\} \Rightarrow \) Painter wins.

\( G = C_3, \mathcal{H} = \{\text{planar 2-degenerate}\} \Rightarrow \) Builder wins.

\( G = C_n, \mathcal{H} = \{\text{planar}\} \Rightarrow \) Builder wins.

\( G = P_n \lor K_1, \mathcal{H} = \{\text{planar}\} \Rightarrow \) Builder wins. (extra idea)

** Conj.** On planar graphs, Builder wins if and only if \( G \) is outerplanar.

Grytczuk–Kierstead–Prałat [2008?]

\( \tilde{r}(G) = \min \{k: \text{Builder wins on graphs with} \leq k \text{ edges}\} \).

\( \tilde{r}(P_n) \leq 4n - 7 \), but for trees it can be quadratic.
Bounded-degree Graphs

**Def.** \( S_k \) = family of graphs with maximum degree \( \leq k \).

\[ \text{ostr}(G) = \min \{ k : \text{Builder wins } (G, S_k) \} . \]
Bounded-degree Graphs

**Def.** $S_k = \text{family of graphs with maximum degree } \leq k$.

$$\text{odr}(G) = \min\{k : \text{Builder wins } (G, S_k)\}.$$

**Thm.** $\text{odr}(G) \leq 3 \iff \text{each component of } G \text{ is a path or each component is a subgraph of } K_{1,3}$. 
Bounded-degree Graphs

Def. \( S_k = \) family of graphs with maximum degree \( \leq k \).

\[ \text{odr}(G) = \min \{ k : \text{Builder wins } (G, S_k) \} . \]

Thm. \( \text{odr}(G) \leq 3 \iff \) each component of \( G \) is a path or each component is a subgraph of \( K_{1,3} \).

Thm. \( \text{odr}(G) \leq 2\Delta(G) - 1 \) when \( G \) is a tree, sharp when \( G \) has adjacent vertices of maximum degree.
Bounded-degree Graphs

**Def.** \( S_k \) = family of graphs with maximum degree \( \leq k \).

\[
\text{odr}(G) = \min\{k : \text{Builder wins } (G, S_k)\}.
\]

**Thm.** \( \text{odr}(G) \leq 3 \iff \) each component of \( G \) is a path or each component is a subgraph of \( K_{1,3} \).

**Thm.** \( \text{odr}(G) \leq 2\Delta(G) - 1 \) when \( G \) is a tree, sharp when \( G \) has adjacent vertices of maximum degree.

**Thm.** \( 4 \leq \text{odr}(C_n) \leq 5 \).
Def. $S_k =$ family of graphs with maximum degree $\leq k$.

$$\text{odr}(G) = \min\{k: \text{Builder wins } (G, S_k)\}.$$  

Thm. $\text{odr}(G) \leq 3 \iff$ each component of $G$ is a path or each component is a subgraph of $K_{1,3}$.

Thm. $\text{odr}(G) \leq 2\Delta(G) - 1$ when $G$ is a tree, sharp when $G$ has adjacent vertices of maximum degree.

Thm. $4 \leq \text{odr}(C_n) \leq 5$.

Thm. $\text{odr}(C_n) = 4$ if $n$ is even or large or 3.
Bounded-degree Graphs

**Def.** $S_k =$ family of graphs with maximum degree $\leq k$.

$$\text{odr}(G) = \min\{k: \text{Builder wins } (G, S_k)\}.$$  

**Thm.** $\text{odr}(G) \leq 3 \iff$ each component of $G$ is a path or each component is a subgraph of $K_{1,3}$.

**Thm.** $\text{odr}(G) \leq 2\Delta(G) - 1$ when $G$ is a tree, sharp when $G$ has adjacent vertices of maximum degree.

**Thm.** $4 \leq \text{odr}(C_n) \leq 5$.

**Thm.** $\text{odr}(C_n) = 4$ if $n$ is even or large or 3.

**Thm.** $\text{odr}(G) \leq 8$ if $\Delta(G) \leq 2$ (maybe less).
Warmup - Paths

**Thm.** $\text{odr}(P_n) \leq 3$.

**Pf.** Stronger statement: In $S_3$, Builder forces a monochr. $P_m$ with $m \geq n$ whose ends have no other edges.
**Thm.** $\text{o}dr(P_n) \leq 3$.

**Pf.** Stronger statement: In $S_3$, Builder forces a monochromatic $P_m$ with $m \geq n$ whose ends have no other edges.

Induction step: Play the $P_{n-1}$-strategy enough times to get $n-2$ such paths in the same color, say red.
**Thm.** \( \text{odr}(P_n) \leq 3. \)

**Pf.** Stronger statement: In \( S_3 \), Builder forces a monochr. \( P_m \) with \( m \geq n \) whose ends have no other edges.

Induction step: Play the \( P_{n-1} \)-strategy enough times to get \( n-2 \) such paths in the same color, say red.

Play a path of \( n-1 \) new edges through their endpoints.
**Thm.** \( \text{odr}(P_n) \leq 3. \)

**Pf.** Stronger statement: In \( S_3 \), Builder forces a monochr. \( P_m \) with \( m \geq n \) whose ends have no other edges.

Induction step: Play the \( P_{n-1} \)-strategy enough times to get \( n - 2 \) such paths in the same color, say red.

Play a path of \( n - 1 \) new edges through their endpoints. If all new edges are blue, done.
Warmup - Paths

**Thm.** \( \text{odr}(P_n) \leq 3. \)

**Pf.** Stronger statement: In \( S_3 \), Builder forces a monochr. \( P_m \) with \( m \geq n \) whose ends have no other edges.

Induction step: Play the \( P_{n-1} \)-strategy enough times to get \( n - 2 \) such paths in the same color, say red.

Play a path of \( n - 1 \) new edges through their endpoints. If all new edges are blue, done. If one is red, done.
Weighted Graphs

- Each vertex $v$ has an “allowed” degree $c(v)$. Weight of $v = $ total number of edges played at $v$. 
Weighted Graphs

- Each vertex $v$ has an “allowed” degree $c(v)$. Weight of $v = \text{total number of edges played at } v$.

**Obs.** If Builder can force a copy of $G$ with weight at most $k$ at each vertex when no restrictions are imposed on edges played, then $\text{odr}(G) \leq k$.

**Pf.** It doesn’t matter what Painter does on an edge with a vertex whose degree exceeds $k$, because this edge cannot lie in the resulting $G$. 

Weighted Graphs

- Each vertex $v$ has an “allowed” degree $c(v)$. Weight of $v$ = total number of edges played at $v$.

**Obs.** If Builder can force a copy of $G$ with weight at most $k$ at each vertex when no restrictions are imposed on edges played, then $\text{odr}(G) \leq k$.

**Pf.** It doesn’t matter what Painter does on an edge with a vertex whose degree exceeds $k$, because this edge cannot lie in the resulting $G$.

**Obs.** If Builder wins $(G, \mathcal{H})$ (weighted $G$), then Builder wins $(mG, \mathcal{H})$.

**Pf.** Play $2m - 1$ times, then pigeonhole on two colors.
**Thm.** Builder can force a weighted $K_{1,m}$ if center has weight $\geq 2m - 1$ or each vertex has weight $\geq m$. 
**Thm.** Builder can force a weighted $K_{1,m}$ if center has weight $\geq 2m - 1$ or each vertex has weight $\geq m$.

**Pf.** With weight $2m - 1$ allowed at center, play $K_{1,2m-1}$. 

---

**Stars**

**Thm.** Builder can force a weighted $K_{1,m}$ if center has weight $\geq 2m - 1$ or each vertex has weight $\geq m$.

**Pf.** With weight $2m - 1$ allowed at center, play $K_{1,2m-1}$. 

---

**Stars**

**Thm.** Builder can force a weighted $K_{1,m}$ if center has weight $\geq 2m - 1$ or each vertex has weight $\geq m$.

**Pf.** With weight $2m - 1$ allowed at center, play $K_{1,2m-1}$. 

---

**Stars**

**Thm.** Builder can force a weighted $K_{1,m}$ if center has weight $\geq 2m - 1$ or each vertex has weight $\geq m$.

**Pf.** With weight $2m - 1$ allowed at center, play $K_{1,2m-1}$. 

---
**Thm.** Builder can force a weighted $K_{1,m}$ if center has weight $\geq 2m - 1$ or each vertex has weight $\geq m$.

**Pf.** With weight $2m - 1$ allowed at center, play $K_{1,2m-1}$. For weight $m$, induct on $m$. Trivial for $m = 1$. 

**Stars**
**Thm.** Builder can force a weighted $K_{1,m}$ if center has weight $\geq 2m - 1$ or each vertex has weight $\geq m$.

**Pf.** With weight $2m - 1$ allowed at center, play $K_{1,2m-1}$. For weight $m$, induct on $m$. Trivial for $m = 1$. For $m > 1$, Builder can force $mK_{1,m-1}$ in $S_{m-1}$.
**Thm.** Builder can force a weighted $K_{1,m}$ if center has weight $\geq 2m - 1$ or each vertex has weight $\geq m$.

**Pf.** With weight $2m - 1$ allowed at center, play $K_{1,2m-1}$.

For weight $m$, induct on $m$. Trivial for $m = 1$.

For $m > 1$, Builder can force $mK_{1,m-1}$ in $S_{m-1}$.

Add $K_{1,m}$ with new center having old centers as leaves.
**Thm.** Builder can force a weighted $K_{1,m}$ if center has weight $\geq 2m - 1$ or each vertex has weight $\geq m$.

**Pf.** With weight $2m - 1$ allowed at center, play $K_{1,2m-1}$. For weight $m$, induct on $m$. Trivial for $m = 1$. For $m > 1$, Builder can force $mK_{1,m-1}$ in $S_{m-1}$. Add $K_{1,m}$ with new center having old centers as leaves.

All blue. One red.
**Thm.** $\text{odr}(C_3) \leq 4$.

**Pf.** If Builder forces a $(4, 2, 2, 2)$-claw, then Builder wins by presenting a triangle on its leaves.
Triangles

**Thm.** $\text{odr}(C_3) \leq 4$.

**Pf.** If **Builder** forces a $(4, 2, 2, 2)$-claw, then **Builder** wins by presenting a triangle on its leaves.

1) Present a claw (centered at $u$), done if monochromatic. Say $uv$ blue, others red.
**Thm.** $\text{o}d\text{r}(C_3) \leq 4$.

**Pf.** If **Builder** forces a $(4, 2, 2, 2)$-claw, then **Builder** wins by presenting a triangle on its leaves.

1) Present a claw (centered at $u$), done if monochromatic. Say $uv$ blue, others red.

2) Present a claw centered at $v$ (gives it degree 4), done if all red. Say $vw$ blue.
**Thm.** $\text{odr}(C_3) \leq 4$.

**Pf.** If Builder forces a $(4, 2, 2, 2)$-claw, then Builder wins by presenting a triangle on its leaves.

1) Present a claw (centered at $u$), done if monochromatic. Say $uv$ blue, others red.

2) Present a claw centered at $v$ (gives it degree 4), done if all red. Say $vw$ blue.

3) Present $uw$. Done if blue; otherwise, red $(4, 2, 2, 2)$-claw.
The Greedy Painter

**Def.** The greedy $\mathcal{F}$-Painter colors each edge red if the resulting red graph lies in $\mathcal{F}$. 
The Greedy Painter

**Def.** The greedy $\mathcal{F}$-Painter colors each edge red if the resulting red graph lies in $\mathcal{F}$.

**Thm.** $S_{m-1}$-Painter makes Builder use $\geq 2m - 1$ at center or $\geq m$ at each vertex to get monochr. $K_{1,m}$. 
The Greedy Painter

**Def.** The greedy $\mathcal{F}$-Painter colors each edge red if the resulting red graph lies in $\mathcal{F}$.

**Thm.** $S_{m-1}$-Painter makes Builder use $\geq 2m - 1$ at center or $\geq m$ at each vertex to get monochr. $K_{1,m}$.

**Pf.** Edge gets blue $\iff$ an endpt already has $m - 1$ red. Painter never makes a red $K_{1,m}$. If a blue $K_{1,m}$, then center has $m - 1$ red or each leaf has $m - 1$ red.
**Def.** The greedy $\mathcal{F}$-Painter colors each edge red if the resulting red graph lies in $\mathcal{F}$.

**Thm.** $S_{m-1}$-Painter makes Builder use $\geq 2m - 1$ at center or $\geq m$ at each vertex to get monochr. $K_{1,m}$.

**Pf.** Edge gets blue if an endpt already has $m - 1$ red. Painter never makes a red $K_{1,m}$. If a blue $K_{1,m}$, then center has $m - 1$ red or each leaf has $m - 1$ red.

**Thm.** $\text{odr}(G) \geq \Delta(G) - 1 + \max_{uv \in E(G)} \min \{d(u), d(v)\}$. 
The Greedy Painter

**Def.** The greedy $\mathcal{F}$-Painter colors each edge red if the resulting red graph lies in $\mathcal{F}$.

**Thm.** $S_{m-1}$-Painter makes Builder use $\geq 2m - 1$ at center or $\geq m$ at each vertex to get monochr. $K_{1,m}$.

**Pf.** Edge gets blue $\iff$ an endpt already has $m-1$ red. Painter never makes a red $K_{1,m}$. If a blue $K_{1,m}$, then center has $m-1$ red or each leaf has $m-1$ red.

**Thm.** $\text{odr}(G) \geq \Delta(G) - 1 + \max_{uv \in E(G)} \min \{ d(u), d(v) \}$.

**Pf.** Let $m = \Delta(G)$. $S_{m-1}$-Painter never makes red $G$. Let $xy$ be an edge with maxmin degree in $G$. A blue $G$ has an edge for $xy$; it has $m-1$ red at one endpt and at least $\min \{ d_G(x), d_G(y) \}$ blue there.
**Theorem.** $\text{odr}(G) \leq 3 \iff$ each component is a path or each component is a subgraph of $K_{1,3}$. 
Small odr

**Thm.** odr($G$) $\leq 3$ $\iff$ each component is a path or each component is a subgraph of $K_{1,3}$.

**Pf.** Suffic.: **Builder** can force long path or $mK_{1,3}$. 
Small odr

Thm. $\text{odr}(G) \leq 3 \iff$ each component is a path or each component is a subgraph of $K_{1,3}$.

Pf. Suffic.: Builder can force long path or $mK_{1,3}$.
Necessity: Builder plays in $S_3$.
Painter uses greedy $\mathcal{L}$, where $\mathcal{L} = \{\text{linear forests}\}$. Red graph is always a linear forest. What of blue?
Small $odr$

**Thm.** $odr(G) \leq 3 \iff$ each component is a path or each component is a subgraph of $K_{1,3}$.

**Pf.** Suffic.: Builder can force long path or $mK_{1,3}$.

Necessity: Builder plays in $S_3$.
Painter uses greedy $\mathcal{L}$, where $\mathcal{L} = \{\text{linear forests}\}$.
Red graph is always a linear forest. What of blue?

To create degree 3 in blue at $v$, each neighbor must already have two red and hence no more blue (in $S_3$).
\[ \therefore \text{blue component is } K_{1,3}. \]
odr \leq 3, \text{ cont.}

Long blue path unforceable against greedy $\mathcal{L}$-Painter.

Suppose blue $\langle u, v, w \rangle$ has another blue at $u$ and $w$ (maybe $uw$). Each of $u, v, w$ has at most one red (in $S_3$).
odr ≤ 3, cont.

Long blue path unforceable against greedy $\mathcal{L}$-Painter.

Suppose blue $\langle u, v, w \rangle$ has another blue at $u$ and $w$ (maybe $uw$). Each of $u, v, w$ has at most one red (in $S_3$).

$uv, vw$ blue $\Rightarrow \exists$ red $u, v$-path and red $v, w$-path.

This violates red $\subseteq \mathcal{L}$. 

![Diagram](attachment:diagram.png)
odr ≤ 3, cont.

Long blue path unforceable against greedy $\mathcal{L}$-Painter.

Suppose blue $\langle u, v, w \rangle$ has another blue at $u$ and $w$ (maybe $uw$). Each of $u, v, w$ has at most one red (in $S_3$).

$uv, vw$ blue $\Rightarrow \exists \text{ red } u, v$-path and red $v, w$-path.

This violates red $\subseteq \mathcal{L}$.

Builder forces blue $K_{1,3} + P_4$ against greedy $\mathcal{L}$-Painter.
odr $\leq 3$, cont.

Long blue path unforceable against greedy $\mathcal{L}$-Painter.

Suppose blue $\langle u, v, w \rangle$ has another blue at $u$ and $w$ (maybe $uw$). Each of $u, v, w$ has at most one red (in $S_3$). $uv, vw$ blue $\Rightarrow \exists$ red $u, v$-path and red $v, w$-path. This violates $\text{red} \subseteq \mathcal{L}$.

Builder forces blue $K_{1,3} + P_4$ against greedy $\mathcal{L}$-Painter.

However, greedy $S_2$-Painter yields 
$\text{odr}(K_{1,3} + P_4) \geq 3 - 1 + \min\{2, 2\} = 4.$
**Cor.** If $G$ has adjacent vertices of maximum degree, then $\text{odr}(G) \geq 2\Delta(G) - 1$. 
Trees

**Cor.** If $G$ has adjacent vertices of maximum degree, then $\text{odr}(G) \geq 2\Delta(G) - 1$.

**Thm.** If $G$ is a tree, then $\text{odr}(G) \leq 2\Delta(G) - 1$. 
Cor. If $G$ has adjacent vertices of maximum degree, then $\text{odr}(G) \geq 2\Delta(G) - 1$.

Thm. If $G$ is a tree, then $\text{odr}(G) \leq 2\Delta(G) - 1$.  
[Stronger result: $\text{odr}(G) \leq d_1 + d_2 - 1$.]
**Cor.** If $G$ has adjacent vertices of maximum degree, then $\text{o dr}(G) \geq 2\Delta(G) - 1$.

**Thm.** If $G$ is a tree, then $\text{o dr}(G) \leq 2\Delta(G) - 1$. [Stronger result: $\text{o dr}(G) \leq d_1 + d_2 - 1$.]

**Pf.** Idea: Builder forces an arbitrarily large monochr. complete $k$-ary tree, where $k = \Delta(G) - 1$. 
**Cor.** If $G$ has adjacent vertices of maximum degree, then $\text{odr}(G) \geq 2\Delta(G) - 1$.

**Thm.** If $G$ is a tree, then $\text{odr}(G) \leq 2\Delta(G) - 1$.

[Stronger result: $\text{odr}(G) \leq d_1 + d_2 - 1$.]

**Pf.** Idea: Builder forces an arbitrarily large monochromatic complete $k$-ary tree, where $k = \Delta(G) - 1$.

Candidate tree $T_R$ or $T_B$ has an active vertex $x_R$ or $x_B$ - a vertex of least depth w/o $k$ children via its own color.
Cor. If $G$ has adjacent vertices of maximum degree, then $\text{odr}(G) \geq 2\Delta(G) - 1$.

Thm. If $G$ is a tree, then $\text{odr}(G) \leq 2\Delta(G) - 1$. [Stronger result: $\text{odr}(G) \leq d_1 + d_2 - 1$.]

Pf. Idea: Builder forces an arbitrarily large monochr. complete $k$-ary tree, where $k = \Delta(G) - 1$.

Candidate tree $T_R$ or $T_B$ has an active vertex $x_R$ or $x_B$ - a vertex of least depth w/o $k$ children via its own color.

Invariant: In $T_R$, each vertex other than $x_R$ either 1) is a leaf in $T_R$ with no other incident edge (wt 1), or 2) has $k$ red children and at most $k$ blue incident edges. (Symmetrically for $T_B$).
Invariant Condition

In $T_R$, each vertex other than $x_R$ either
1) is a leaf in $T_R$ with no other incident edge (wt 1), or
2) has $k$ red children and at most $k$ blue incident edges.
(Symmetrically for $T_B$).
Invariant Condition

In $T_R$, each vertex other than $x_R$ either
1) is a leaf in $T_R$ with no other incident edge (wt 1), or
2) has $k$ red children and at most $k$ blue incident edges.
(Symmetrically for $T_B$).

An active vertex is
satisfied if it has $k$ children via its own color.
dangerous if it has $k$ incident edges of the other color.
Builder strategy

Builder plays pendant edges at active vertices (in $T_R$ or $T_B$) until Painter makes one satisfied or dangerous.
Builder Strategy

**Builder** plays pendant edges at active vertices (in $T_R$ or $T_B$) until **Painter** makes one satisfied or dangerous.

When an active vertex is satisfied, **Builder** rechooses it (closest to root w/o $k$ children via its own color).
**Builder Strategy**

**Builder** plays pendant edges at active vertices (in $T_R$ or $T_B$) until **Painter** makes one satisfied or dangerous.

When an active vertex is satisfied, **Builder** rechooses it (closest to root w/o $k$ children via its own color).

If $x_R$ and $x_B$ are both dangerous,

![Diagram of trees $T_R$ and $T_B$ with vertices $x_R$, $x_B$]
Builder Strategy

Builder plays pendant edges at active vertices (in $T_R$ or $T_B$) until Painter makes one satisfied or dangerous.

When an active vertex is satisfied, Builder rechooses it (closest to root w/o $k$ children via its own color).

If $x_R$ and $x_B$ are both dangerous, Builder plays $x_Rx_B$. 
Builder Strategy

Builder plays pendant edges at active vertices (in $T_R$ or $T_B$) until Painter makes one satisfied or dangerous.

When an active vertex is satisfied, Builder rechooses it (closest to root w/o $k$ children via its own color).

If $x_R$ and $x_B$ are both dangerous, Builder plays $x_Rx_B$.

This edge enters the tree for its color, dragging the other tree with it.
Builder Strategy

Builder plays pendant edges at active vertices (in $T_R$ or $T_B$) until Painter makes one satisfied or dangerous.

When an active vertex is satisfied, Builder rechooses it (closest to root w/o $k$ children via its own color).

If $x_R$ and $x_B$ are both dangerous, Builder plays $x_R x_B$.

This edge enters the tree for its color, dragging the other tree with it.

Then Builder regenerates the other tree.
The Consistent Painter

**Def.** Painter follows a **consistent** strategy if the color given to a newly presented edge depends only on the current 2-colored components containing its endpoints (regardless of what has been played elsewhere).
The Consistent Painter

**Def.** Painter follows a **consistent** strategy if the color given to a newly presented edge depends only on the current 2-colored components containing its endpoints (regardless of what has been played elsewhere).

**Thm.** If $\mathcal{H}$ is an additive family (closed under disjoint unions), and $A$ is a Painter strategy on $\mathcal{H}$, then there is a consistent Painter strategy $A'$ on $\mathcal{H}$ such that for any list $E'$ presented by Builder, there is another list $E$ such that $A'(E') \subseteq A(E)$ (as 2-colored graphs).
The Consistent Painter

**Def.** Painter follows a consistent strategy if the color given to a newly presented edge depends only on the current 2-colored components containing its endpoints (regardless of what has been played elsewhere).

**Thm.** If $\mathcal{H}$ is an additive family (closed under disjoint unions), and $A$ is a Painter strategy on $\mathcal{H}$, then there is a consistent Painter strategy $A'$ on $\mathcal{H}$ such that for any list $E'$ presented by Builder, there is another list $E$ such that $A'(E') \subseteq A(E)$ (as 2-colored graphs).

**Cor.** To prove that $\text{odr}(G) \leq k$, it suffices to show that Builder can win against any consistent Painter on $S_k$. 
The Consistent Painter

**Def.** Painter follows a consistent strategy if the color given to a newly presented edge depends only on the current 2-colored components containing its endpoints (regardless of what has been played elsewhere).

**Thm.** If $\mathcal{H}$ is an additive family (closed under disjoint unions), and $\mathcal{A}$ is a Painter strategy on $\mathcal{H}$, then there is a consistent Painter strategy $\mathcal{A}'$ on $\mathcal{H}$ such that for any list $E'$ presented by Builder, there is another list $E$ such that $\mathcal{A}'(E') \subseteq \mathcal{A}(E)$ (as 2-colored graphs).

**Cor.** To prove that $\text{odr}(G) \leq k$, it suffices to show that Builder can win against any consistent Painter on $S_k$.

**Pf.** $\forall$ Painter $\mathcal{A}$, $\exists$ consistent Painter $\mathcal{A}'$ s.t. Builder can force against $\mathcal{A}$ whatever he can force against $\mathcal{A}'$. 
Assume Builder plays on $S_k$ and Painter is consistent.

**Lem.** Let $F_1, F_2$ be weighted graphs Builder can force in red, with vertices $u_1, u_2$. Form $F$ from $F_1 + F_2$ by adding $u_1 u_2$ and increasing weights on $u_1$ and $u_2$ by 2. If $q$ is even, then Builder can force a red $F$ or a blue $C_q$. 
Even Cycles

Assume Builder plays on $S_k$ and Painter is consistent.

Lem. Let $F_1, F_2$ be weighted graphs Builder can force in red, with vertices $u_1, u_2$. Form $F$ from $F_1 + F_2$ by adding $u_1u_2$ and increasing weights on $u_1$ and $u_2$ by 2. If $q$ is even, then Builder can force a red $F$ or a blue $C_q$.

Pf. Builder forces $q/2$ copies of $F_1$ and $F_2$ and then adds a cycle alternating between the copies of $u_1$ and $u_2$. ■
Consistent Painter makes the same monochromatic $P_3$ (with weights 2) in any isolated triangle; we may assume it is red. Painter wants to avoid a monochromatic $C_q$. 
Consistent Painter makes the same monochromatic $P_3$ (with weights 2) in any isolated triangle; we may assume it is red. Painter wants to avoid a monochromatic $C_q$. 
Trees for Even Cycles

Consistent Painter makes the same monochr. $P_3$ (with weights 2) in any isolated triangle; we may assume it is red. Painter wants to avoid a monochromatic $C_q$. 

$q = 4$
Consistent Painter makes the same monochromatic $P_3$ (with weights 2) in any isolated triangle; we may assume it is red. Painter wants to avoid a monochromatic $C_q$.

$q = 8$
Consistent Painter makes the same monochromatic $P_3$ (with weights 2) in any isolated triangle; we may assume it is red. Painter wants to avoid a monochromatic $C_q$.

Extension by $P_3$ lengthens both sides by 2.
Extension by $K_1$ lengthens both sides by 1.
Special Case: $C_6$

Consistent Painter makes consistent triangles.

Case 1: monochromatic
Special Case: \( C_6 \)

Consistent Painter makes consistent triangles.

Case 1: monochromatic
Special Case: $C_6$

Consistent Painter makes consistent triangles.

Case 1: monochromatic
Special Case: $C_6$

Consistent Painter makes consistent triangles.

Case 1: monochromatic

Case 2: not monochromatic
Special Case: $C_6$

Consistent Painter makes consistent triangles.

Case 1: monochromatic

Case 2: not monochromatic
Special Case: $C_6$

Consistent Painter makes consistent triangles.

Case 1: monochromatic

Case 2: not monochromatic
Summary for Cycles

$C_{10}$ done similarly.

**Thm.** For even $n$ with $n \geq 4$, $\text{odr}(C_n) = 4$. 
Summary for Cycles

$C_{10}$ done similarly.

**Thm.** For even $n$ with $n \geq 4$, $\text{odr}(C_n) = 4$.

Recall $\text{odr}(C_3) = 4$ by claw + triangle.
Summary for Cycles

\( C_{10} \) done similarly.

**Thm.** For even \( n \) with \( n \geq 4 \), \( \text{odr}(C_n) = 4 \).

Recall \( \text{odr}(C_3) = 4 \) by claw + triangle.

**Thm.** For all \( n \), \( \text{odr}(C_n) \leq 5 \).
Summary for Cycles

\( C_{10} \) done similarly.

**Thm.** For even \( n \) with \( n \geq 4 \), \( \text{odr}(C_n) = 4 \).

Recall \( \text{odr}(C_3) = 4 \) by claw + triangle.

**Thm.** For all \( n \), \( \text{odr}(C_n) \leq 5 \).

**Thm.** If \( 337 \leq n \leq 514 \) or \( n \geq 689 \), then \( \text{odr}(C_n) = 4 \).
Summary for Cycles

$C_{10}$ done similarly.

**Thm.** For even $n$ with $n \geq 4$, $\text{odr}(C_n) = 4$.

Recall $\text{odr}(C_3) = 4$ by claw + triangle.

**Thm.** For all $n$, $\text{odr}(C_n) \leq 5$.

**Thm.** If $337 \leq n \leq 514$ or $n \geq 689$, then $\text{odr}(C_n) = 4$.

**Ques.** Can Builder win every $(C_n, S_4)$?
Summary for Cycles

$C_{10}$ done similarly.

**Thm.** For even $n$ with $n \geq 4$, $\text{odr}(C_n) = 4$.

Recall $\text{odr}(C_3) = 4$ by claw + triangle.

**Thm.** For all $n$, $\text{odr}(C_n) \leq 5$.

**Thm.** If $337 \leq n \leq 514$ or $n \geq 689$, then $\text{odr}(C_n) = 4$.

**Ques.** Can Builder win every $(C_n, S_4)$?

**Ques.** Does $\text{odr}(C_5)$ equal 4? (otherwise 5)
**Lem.** On $S_k$ with $k \geq 4$ against a consistent Painter, if Builder can force red $F$ or monochr. $C_q$ ($q$ odd), then Builder can force a red $F'$ or a monochr. $C_q$, where $F' = \text{add pendant } uv$, increasing $c(u)$ by 2, set $c(v) = 2$. 
Odd Cycles

**Lem.** On $S_k$ with $k \geq 4$ against a consistent Painter, if Builder can force red $F$ or monochr. $C_q$ ($q$ odd), then Builder can force a red $F'$ or a monochr. $C_q$, where $F' = \text{add pendant } uv$, increasing $c(u)$ by 2, set $c(v) = 2$.

**Thm.** $\text{odr}(C_q) \leq 5$ when $q$ is odd.
**Lem.** On $S_k$ with $k \geq 4$ against a consistent Painter, if Builder can force red $F$ or monochr. $C_q$ ($q$ odd), then Builder can force a red $F'$ or a monochr. $C_q$, where $F' = \text{add pendant } uv$, increasing $c(u)$ by 2, set $c(v) = 2$.

**Thm.** $\text{odr}(C_q) \leq 5$ when $q$ is odd.

**Pf.** Force monochr. $P_q$ (say red) with weights 3.

\begin{center}
\begin{tikzpicture}
    \draw[red] (0,0) -- (11,0);
    \foreach \x in {0,1,2,3,4,5,6,7,8,9,10,11}
    \draw[fill=black] (\x,0) circle (2pt);
    \foreach \x in {0,1,2,3,4,5,6,7,8,9,10,11}
    \draw[red,thick] (\x,0) -- (\x+1,0);
    \end{tikzpicture}
\end{center}

\[3 \ 3 \ 3 \ 3 \ 3 \ 3 \ 3 \ 3 \ 3 \ 3 \]
Odd Cycles

**Lem.** On $S_k$ with $k \geq 4$ against a consistent Painter, if Builder can force red $F$ or monochr. $C_q$ ($q$ odd), then Builder can force a red $F'$ or a monochr. $C_q$, where $F' = \text{add pendant } uv$, increasing $c(u)$ by 2, set $c(v) = 2$.

**Thm.** $\text{oord}(C_q) \leq 5$ when $q$ is odd.

**Pf.** Force monochr. $P_q$ (say red) with weights 3. Grow pendant paths.

```
3 5 5 5 5 5 5 5 3
2 2 2 2 2 2 2
```
Odd Cycles

**Lem.** On $S_k$ with $k \geq 4$ against a consistent Painter, if Builder can force red $F$ or monochr. $C_q$ ($q$ odd), then Builder can force a red $F'$ or a monochr. $C_q$, where $F' = \text{add pendant } uv$, increasing $c(u)$ by 2, set $c(v) = 2$.

**Thm.** $\text{odr}(C_q) \leq 5$ when $q$ is odd.

**Pf.** Force monochr. $P_q$ (say red) with weights 3. Grow pendant paths.

Leaf distances $q - 1$ (opposite halves or to middle). Cycle through the leaves is all blue or some red.
Ques. Does there exist a function $f$ such that $\text{odr}(G) \leq f(\Delta(G))$ for every graph $G$?
Other Open Questions

**Ques.** Does there exist a function $f$ such that $\text{odr}(G) \leq f(\Delta(G))$ for every graph $G$?

**Ques.** What is $\text{odr}(C_4 + e)$? (In $\{5, 6, 7\}$.)

What is $\text{odr}(K_{1,3} + e)$? (In $\{4, 5\}$.)

Lower bounds by maxmin degree argument.
Upper bounds by **Builder** strategy and case analysis.
Other Open Questions

**Ques.** Does there exist a function $f$ such that $\text{odr}(G) \leq f(\Delta(G))$ for every graph $G$?

**Ques.** What is $\text{odr}(C_4 + e)$? (In \{5, 6, 7\}.)
What is $\text{odr}(K_{1,3} + e)$? (In \{4, 5\}.)

Lower bounds by maxmin degree argument.
Upper bounds by **Builder** strategy and case analysis.

**Ques.** For the double-star tree $G_{a,b}$ with central vertices of degrees $a$ and $b$, the maxmin degree argument yields $\text{odr}(G_{a,b}) \geq a + b - 1$. Equality? Yes!
Other Open Questions

**Ques.** Does there exist a function $f$ such that $\text{odr}(G) \leq f(\Delta(G))$ for every graph $G$?

**Ques.** What is $\text{odr}(C_4 + e)$? (In $\{5, 6, 7\}$.)
What is $\text{odr}(K_{1,3} + e)$? (In $\{4, 5\}$.)

Lower bounds by maxmin degree argument.
Upper bounds by **Builder** strategy and case analysis.

**Ques.** For the double-star tree $G_{a,b}$ with central vertices of degrees $a$ and $b$, the maxmin degree argument yields $\text{odr}(G_{a,b}) \geq a + b - 1$. Equality? Yes!

**Thm.** If a tree $T$ with $\Delta(T) = a$ has exactly one vertex of degree $a$, and all others have degree at most $b$ (where $b \leq a$), then $\text{odr}(T) \leq a + b - 1$. 

A Different Problem

Harder for Builder: play in $S_k$, but all edges at once.
A Different Problem

Harder for Builder: play in $S_k$, but all edges at once.

**Def.** degree Ramsey number $dr(G)$ (Milans)

$dr(G) = \min \{ k : H \to G \text{ for some } H \in S_k \}$. 
A Different Problem

Harder for **Builder**: play in $S_k$, but all edges at once.

**Def.** degree Ramsey number $d_r(G)$ (Milans)

$\displaystyle d_r(G) = \min \{ k : H \rightarrow G \text{ for some } H \in S_k \}$.

**Obs.** $\text{odr}(G) \leq d_r(G) < R(G, G)$. 
A Different Problem

Harder for Builder: play in $S_k$, but all edges at once.

**Def.** degree Ramsey number $dr(G)$ (Milans)

\[ dr(G) = \min \{ k : H \to G \text{ for some } H \in S_k \}. \]

**Obs.** $odr(G) \leq dr(G) < R(G,G)$.

More general problem: allow more colors.

\[ dr(G; s) = \min \{ k : H \overset{s}{\to} G \text{ for some } H \in S_k \}. \]
A Different Problem

Harder for Builder: play in $S_k$, but all edges at once.

**Def.** degree Ramsey number $\text{dr}(G)$ (Milans)

$\text{dr}(G) = \min\{k : H \to G \text{ for some } H \in S_k\}$.

**Obs.** $\text{odr}(G) \leq \text{dr}(G) < R(G, G)$.

More general problem: allow more colors.

$\text{dr}(G; s) = \min\{k : H \overset{s}{\to} G \text{ for some } H \in S_k\}$.

Like $\text{odr}(G)$, some results are known for $\text{dr}(G)$ when $G$ is a path, star, cycle, or tree.
Degree Ramsey for Paths

**Thm.** Alon–Ding–Oporowski–Vertigan[2003] \( dr(P_n; s) \leq 2s \).
Degree Ramsey for Paths

**Thm.** Alon–Ding–Oporowski–Vertigan[2003] \( dr(P_n; s) \leq 2s \).

**Pf.** Let \( H \) be \( 2s \)-regular with girth \( \geq n \). Let \( m = |V(H)| \).
**Thm.** Alon–Ding–Oporowski–Vertigan[2003] \( \text{dr}(P_n; s) \leq 2s \).

**Pf.** Let \( H \) be 2s-regular with girth \( \geq n \). Let \( m = |V(H)| \).

\( s \)-coloring \( sm \) edges puts \( \geq m \) in some color class. Since \( |V(H)| = m \), this subgraph has a cycle.

Since \( H \) has girth \( \geq n \), this color class contains \( P_n \). ■
**Thm.** Alon–Ding–Oporowski–Vertigan [2003] $\text{dr}(P_n; s) \leq 2s$.

**Pf.** Let $H$ be $2s$-regular with girth $\geq n$. Let $m = |V(H)|$. $s$-coloring $sm$ edges puts $\geq m$ in some color class. Since $|V(H)| = m$, this subgraph has a cycle. Since $H$ has girth $\geq n$, this color class contains $P_n$.

**Thm.** Thomassen [1999] Every 3-regular graph has a 2-edge-coloring such that each monochromatic component is contained in $P_6$. 

Degree Ramsey for Paths

**Thm.** Alon–Ding–Oporowski–Vertigan [2003] \( \text{dr}(P_n; s) \leq 2s \).

**Pf.** Let \( H \) be \( 2s \)-regular with girth \( \geq n \). Let \( m = |V(H)| \).

An \( s \)-coloring of \( sm \) edges puts \( \geq m \) in some color class. Since \( |V(H)| = m \), this subgraph has a cycle. Since \( H \) has girth \( \geq n \), this color class contains \( P_n \). 

**Thm.** Thomassen [1999] Every \( 3 \)-regular graph has a 2-edge-coloring such that each monochromatic component is contained in \( P_6 \).

**Cor.** \( \text{dr}(P_n; 2) = 4 \) and \( \text{dr}(P_n; s) \geq 3s/2 \) for \( n > 6 \).
**Thm.** Alon–Ding–Oporowski–Vertigan [2003] $\text{dr}(P_n; s) \leq 2s$.

**Pf.** Let $H$ be $2s$-regular with girth $\geq n$. Let $m = |V(H)|$. $s$-coloring $sm$ edges puts $\geq m$ in some color class. Since $|V(H)| = m$, this subgraph has a cycle. Since $H$ has girth $\geq n$, this color class contains $P_n$.

**Thm.** Thomassen [1999] Every $3$-regular graph has a $2$-edge-coloring such that each monochromatic component is contained in $P_6$.

**Cor.** $\text{dr}(P_n; 2) = 4$ and $\text{dr}(P_n; s) \geq 3s/2$ for $n > 6$.

**Pf.** If $\Delta(H) < 3s/2$, then $H$ is $3s/2$-edge-colorable. Triples of colors form subgraphs with maxdeg $3$. Use two colors on each subgraph.
Degree Ramsey for Cycles

**Thm.** If $G$ is an odd cycle, then $dr(G) \geq 5$. 
Thm. If $G$ is an odd cycle, then $\text{dr}(G) \geq 5$.

Pf. $K_5$ does not arrow any fixed cycle.
Degree Ramsey for Cycles

**Thm.** If $G$ is an odd cycle, then $\text{dr}(G) \geq 5$.

**Pf.** $K_5$ does not arrow any fixed cycle. By Brooks’ Theorem, every connected $H$ in $S_4$ (other than $K_5$) has a proper 4-coloring $f$. 
**Thm.** If $G$ is an odd cycle, then $\text{dr}(G) \geq 5$.

**Pf.** $K_5$ does not arrow any fixed cycle. By Brooks’ Theorem, every connected $H$ in $S_4$ (other than $K_5$) has a proper 4-coloring $f$. Painter 2-colors $uv$ “odd” or “even” by whether $f(u) - f(v)$ is odd or even. Both classes bipartite. ■
Degree Ramsey for Cycles

**Thm.** If $G$ is an odd cycle, then $\text{dr}(G) \geq 5$.

**Pf.** $K_5$ does not arrow any fixed cycle. By Brooks’ Theorem, every connected $H$ in $S_4$ (other than $K_5$) has a proper 4-coloring $f$. Painter 2-colors $uv$ “odd” or “even” by whether $f(u) - f(v)$ is odd or even. Both classes bipartite. ■

**Cor.** $\text{dr}(C_3) = 5$. 
Degree Ramsey for Cycles

**Thm.** If $G$ is an odd cycle, then $dr(G) \geq 5$.

**Pf.** $K_5$ does not arrow any fixed cycle. By Brooks’ Theorem, every connected $H$ in $S_4$ (other than $K_5$) has a proper 4-coloring $f$. Painter 2-colors $uv$ “odd” or “even” by whether $f(u) - f(v)$ is odd or even. Both classes bipartite.

**Cor.** $dr(C_3) = 5$.

**Ques.** Is $dr(C_n)$ bounded? (Maybe always 5?)
**Thm.** If $G$ is an odd cycle, then $\text{dr}(G) \geq 5$.

**Pf.** $K_5$ does not arrow any fixed cycle. By Brooks’ Theorem, every connected $H$ in $S_4$ (other than $K_5$) has a proper 4-coloring $f$. Painter 2-colors $uv$ “odd” or “even” by whether $f(u) - f(v)$ is odd or even. Both classes bipartite.

**Cor.** $\text{dr}(C_3) = 5$.

**Ques.** Is $\text{dr}(C_n)$ bounded? (Maybe always 5?)

- If $G$ is connected and $\text{dr}(G) \leq 4$, then $G$ is a path or an even cycle.
Thm. If $\Delta(G) \geq 2k + 1$, then $dr(G) \geq 4k + 1$. 
Degree Ramsey by Maximum Degree

**Thm.** If $\Delta(G) \geq 2k + 1$, then $\text{dr}(G) \geq 4k + 1$.

**Pf.** If $\Delta(H) \leq 4k$, then $H$ lies in a $4k$-regular graph $H'$. By Petersen’s Theorem, $H'$ decomposes into $2k$ spanning 2-regular subgraphs. Put $k$ in each color to avoid degree $2k + 1$ in one color at any vertex.
Degree Ramsey by Maximum Degree

**Thm.** If $\Delta(G) \geq 2k + 1$, then $dr(G) \geq 4k + 1$.

**Pf.** If $\Delta(H) \leq 4k$, then $H$ lies in a $4k$-regular graph $H'$. By Petersen’s Theorem, $H'$ decomposes into $2k$ spanning 2-regular subgraphs. Put $k$ in each color to avoid degree $2k + 1$ in one color at any vertex.

**Cor.** $dr(K_{1,2k+1};s) = 2sk + 1$.

**Pf.** Upper Bound: $K_{1,2sk+1} \xrightarrow{s} K_{1,2k+1}$. 


Degree Ramsey by Maximum Degree

**Thm.** If $\Delta(G) \geq 2k + 1$, then $\text{dr}(G) \geq 4k + 1$.

**Pf.** If $\Delta(H) \leq 4k$, then $H$ lies in a $4k$-regular graph $H'$. By Petersen’s Theorem, $H'$ decomposes into $2k$ spanning 2-regular subgraphs. Put $k$ in each color to avoid degree $2k + 1$ in one color at any vertex. □

**Cor.** $\text{dr}(K_{1,2k+1}; s) = 2sk + 1$.

**Pf.** Upper Bound: $K_{1,2sk+1} \rightarrow_s K_{1,2k+1}$. □

**Thm.** $\text{dr}(K_{1,2k}; s) \geq 2sk - s$, with equality for even $s$.
Degree Ramsey by Maximum Degree

**Thm.** If $\Delta(G) \geq 2k + 1$, then $dr(G) \geq 4k + 1$.

**Pf.** If $\Delta(H) \leq 4k$, then $H$ lies in a $4k$-regular graph $H'$. By Petersen’s Theorem, $H'$ decomposes into $2k$ spanning 2-regular subgraphs. Put $k$ in each color to avoid degree $2k + 1$ in one color at any vertex.

**Cor.** $dr(K_{1,2k+1}; s) = 2sk + 1$.

**Pf.** Upper Bound: $K_{1,2sk+1} \rightarrow_s K_{1,2k+1}$.

**Thm.** $dr(K_{1,2k}; s) \geq 2sk - s$, with equality for even $s$.

**Pf.** Lower Bound: By Vizing’s Theorem, $\Delta(G) = 2sk - s - 1 \Rightarrow G$ is $s(2k - 1)$-edge-colorable. Put $2k - 1$ of these matchings into each color.
**Degree Ramsey by Maximum Degree**

**Thm.** If \( \Delta(G) \geq 2k + 1 \), then \( \text{dr}(G) \geq 4k + 1 \).

**Pf.** If \( \Delta(H) \leq 4k \), then \( H \) lies in a \( 4k \)-regular graph \( H' \). By Petersen’s Theorem, \( H' \) decomposes into \( 2k \) spanning \( 2 \)-regular subgraphs. Put \( k \) in each color to avoid degree \( 2k + 1 \) in one color at any vertex.

**Cor.** \( \text{dr}(K_{1,2k+1}; s) = 2sk + 1 \).

**Pf.** Upper Bound: \( K_{1,2sk+1} \xrightarrow{s} K_{1,2k+1} \).

**Thm.** \( \text{dr}(K_{1,2k}; s) \geq 2sk - s \), with equality for even \( s \).

**Pf.** Lower Bound: By Vizing’s Theorem, 
\( \Delta(G) = 2sk - s - 1 \) \( \Rightarrow \) \( G \) is \( s(2k - 1) \)-edge-colorable. 
Put \( 2k - 1 \) of these matchings into each color.

**Upper Bound:** If \( G \) is \( s(2k - 1) \)-regular with odd order, then avoiding \( K_{1,2k} \) puts degree \( 2k - 1 \) in each color at each vertex: regular \( w \) odd-degree & odd order.
Thm. (Jiang) If $G$ is a tree, then $dr(G; s) \leq 2s\Delta(G)$. 
**Thm.** (Jiang) If $G$ is a tree, then $\text{dr}(G; s) \leq 2s\Delta(G)$.

**Pf.** Let $H$ be $2s\Delta(G)$-regular with girth $> \text{diam}(G)$. 
Thm. (Jiang) If $G$ is a tree, then $dr(G; s) \leq 2s\Delta(G)$.

Pf. Let $H$ be $2s\Delta(G)$-regular with girth $> \text{diam}(G)$. $s$-coloring $E(H)$ yields $\text{avgdeg} \geq 2\Delta(G)$ in some color. It has a subgraph with minimum degree $\geq \Delta(G)$. 
Thm. (Jiang) If $G$ is a tree, then $\text{dr}(G; s) \leq 2s\Delta(G)$.

Pf. Let $H$ be $2s\Delta(G)$-regular with girth $> \text{diam}(G)$. $s$-coloring $E(H)$ yields $\text{avgdeg} \geq 2\Delta(G)$ in some color. It has a subgraph with minimum degree $\geq \Delta(G)$. In this color find $G$, since each vertex has $\Delta(G) - 1$ new children until depth $\text{radius}(G)$ is reached.
**Thm.** (Jiang) If $G$ is a tree, then $\text{dr}(G; s) \leq 2s\Delta(G)$.

**Pf.** Let $H$ be $2s\Delta(G)$-regular with girth $> \text{diam}(G)$. $s$-coloring $E(H)$ yields $\text{avgdeg} \geq 2\Delta(G)$ in some color. It has a subgraph with minimum degree $\geq \Delta(G)$. In this color find $G$, since each vertex has $\Delta(G) - 1$ new children until depth $\text{radius}(G)$ is reached.

**Ques.** Is $\text{dr}(G)$ bounded by a function of $\Delta(G)$? (Would imply the analogue for $\text{odr}(G)$.)
Thm. (Jiang) If $G$ is a tree, then $\text{dr}(G; s) \leq 2s\Delta(G)$.

Pf. Let $H$ be $2s\Delta(G)$-regular with girth $> \text{diam}(G)$. $s$-coloring $E(H)$ yields avgdeg $\geq 2\Delta(G)$ in some color. It has a subgraph with minimum degree $\geq \Delta(G)$. In this color find $G$, since each vertex has $\Delta(G) - 1$ new children until depth radius($G$) is reached.

Ques. Is $\text{dr}(G)$ bounded by a function of $\Delta(G)$? (Would imply the analogue for $\text{odr}(G)$.)

Ques. Is $\text{dr}(G)$ bounded by a function of $\text{o dr}(G)$? (We still have no graph with $\text{dr}(G) > \text{o dr}(G) + 1$.)
**Thm.** (Jiang) If $G$ is a tree, then $dr(G; s) \leq 2s\Delta(G)$.

**Pf.** Let $H$ be $2s\Delta(G)$-regular with girth $> \text{diam}(G)$. $s$-coloring $E(H)$ yields $\text{avgdeg} \geq 2\Delta(G)$ in some color. It has a subgraph with minimum degree $\geq \Delta(G)$. In this color find $G$, since each vertex has $\Delta(G) - 1$ new children until depth $\text{radius}(G)$ is reached.

**Ques.** Is $dr(G)$ bounded by a function of $\Delta(G)$? (Would imply the analogue for $odr(G)$.)

**Ques.** Is $dr(G)$ bounded by a function of $odr(G)$? (We still have no graph with $dr(G) > odr(G) + 1$.)

**Ques.** What is $dr(C_n)$? What is $dr(K_{1,2k}; s)$ when $s$ is odd?