Greedy Algorithms

For Zach Chroman

Evan Chen

March 21, 2016

SL 2004 A6

Problem

$f(x^2+y^2+2f(xy)) = f(x+y)^2.$

By placing $y=0$

$$f(x^2+2f(0)) = f(x)^2$$

so now we can put

$$f(x^2+y^2+2f(xy)) = f(x^2+y^2+2xy+2f(0)).$$

If $a = x+y$, $b = xy$ then this is

$$f(a^2-2b+2f(b)) = f(a^2+2f(0)) \qquad a^2 \ge 4b.$$

So suppose there exists a b such that $2f(b)-2b \neq 2f(0)$.

- Either the case where $f(x) = x + *$, get $f(x) = x$.
- f is periodic for large x.
  - Do work to narrow to done f constant for $x \gg 0$.
  - Can show $f(x) \in \{0, \pm 1\}$ for every x.

$f(x) = x$, $f(x) = 0$, and

$$f(x) = \begin{cases} +1 & x \notin S \\ -1 & x \in S \end{cases}$$

where $S \subseteq (-\infty, -\frac{2}{3})$ is arbitrary.

Global vs. Local

- Global methods: double-counting, linearity of expectation, etc.
  Graph metaphor: $\sum \deg v = 2E$.
- Local methods: greedy algorithms.
  Graph metaphor: start at a vertex and start walking.

Greedy algorithm: you have a search space.

IMO 2014 Problem 6

Example (IMO 2014/6)

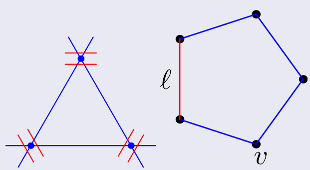
Prove that for all sufficiently large n, in any set of n lines in general position it is possible to colour at least $\sqrt{n}$ lines blue in such a way that none of its finite regions has a completely blue boundary.

Strategy

Color lines blue until stuck.

Proof this strategy works.

Look at a maximal configuration. Claim that in here, at least $\sqrt{n}$ lines are blue.



So suppose there k blue lines and $n-k$ red lines. Then there are $\binom{k}{2}$ intersections of two blue lines. Moreover every red line is part of an almost-blue polygon. So can associate every red line to a blue intersection.

By "geometry", at most two red lines per blue vertex. Thus

$$\binom{k}{2} \geq \frac{1}{2}(n-k) \implies k \geq \sqrt{n}.$$

□

Putnam example

Example (Putnam 1979)

In the plane are n red points and n blue points, no three collinear. Prove we can join them with n segments, each joining a red point to a blue point, such that no two segments intersect.

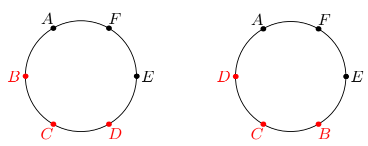
- Idea: do a "greedy" algorithm
- Start anywhere, and then break intersections.
- Experiment: this algorithm eventually terminates at a good state.
- Number of intersections is not a valid monovariant here.
- Sum of distances works as monovariant.

Dirac

Example (Dirac's Theorem)

Show that any graph on n vertices, where each vertex has degree at least $n/2$, has a Hamiltonian cycle.

- Start by arranging the vertices in a circle arbitrarily.
- Say a pair of adjacent vertices is bad if the two vertices are not neighbors of each other.
- Given a situation where there is at least one bad pair is it possible to decrease the number of bad pairs?
- Reflect a block of people: only disrupt two pairs.



Suppose DE is a bad pair. Then want to find a person B such that both DA and EB good.
This is possible by Pigeonhole.

PUMaC Finals

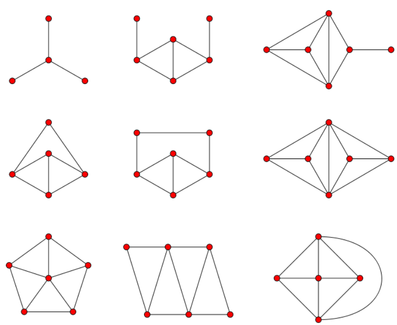
Problem (PUMaC Finals)

Let G be a graph and let k be a positive integer. A k-star is a set of k edges with a common endpoint and a k-matching is a set of k edges such that no two have a common endpoint. Prove that if G has more than $2(k-1)^2$ edges then it either has a k-star or a k-matching.

Line graph: suffices to show either K_k or empty graph on k vertices as a induced subgraph. Line graph has $\geq 2(k-1)^2$ vertices, but we need some bound on number of edges. It would suffice if

$$R(k,k) \leq 2(k-1)^2$$

but this is not true at all.
Need some condition on line graph: line graphs are claw-free, for example.
Complete list of forbidden induced subgraphs:



Problem (PUMaC Finals)

Let G be a graph and let k be a positive integer. A k-star is a set of k edges with a common endpoint and a k-matching is a set of k edges such that no two have a common endpoint. Prove that if G has more than $2(k-1)^2$ edges then it either has a k-star or a k-matching.

Assume G has more than $2(k-1)^2$ edges but has all degrees $\le k-1$.
Take a maximal matching ($\iff$ greedy grab edges). In each edge (v, w) in the matching, you have at most

$$1 + 2(2k-2) = 2k-3$$

edges touching one of v, w.
Therefore, total number of edges is at most $(k-1)(2k-3) < 2(k-1)^2$.
(In general: if looking for big matching, maximal matching is a very good thing to consider.)