CSCI 3100

Approximation alg (pt 2)

Announcements

- Make sure you have a (Friday) spot for oral grading
- In office: 1-2, 3-4 tomorrow 9-10

Dfns for Approx:

Let $OPT(x)$ = value of optimal solution

$A(x)$ = value of solution computed by algorithm A

A is an $\alpha(n)$-approximation algorithm if:

$$\left.\begin{array}{l} \dfrac{OPT(x)}{A(x)} \le \alpha(n) \\[2em] \text{and} \quad \dfrac{A(x)}{OPT(x)} \le \alpha(n) \end{array}\right\}$$

max vs min

- $\alpha(n)$ is called the approximation factor.

Last time

Greedy load balancing;
minimizing makespan

Result;

greedy alg (X)

$\leq 2\,\mathrm{OPT}(X)$

offline (inputs in sorted order)

$\leq \frac{3}{2}\,\mathrm{OPT}(X)$

Vertex Cover

NP-Hard.

Shall we try greedy again?

How should we be greedy?

While edges remain

Take max degree vertex v & add it to our cover C.

Delete v & all adjacent edges

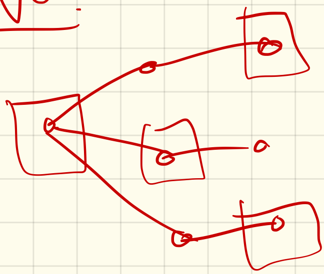
(assuming connected graph)

Algorithm:

```
GREEDYVERTEXCOVER(G):
  C ← ∅
  while G has at least one edge
      v ← vertex in G with maximum degree
      G ← G \ v
      C ← C ∪ v
  return C
```

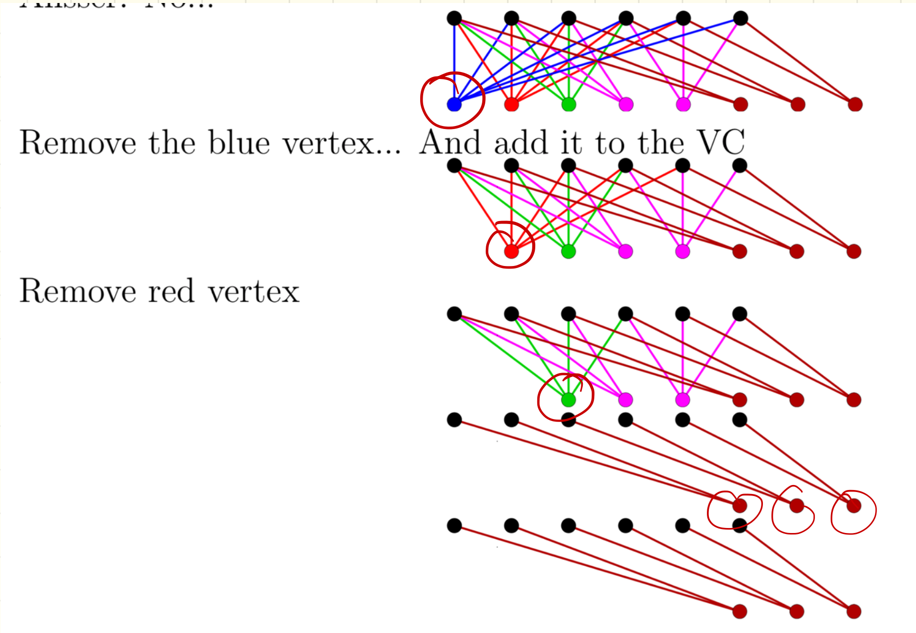
Question: Is this always optimal?

No:



Q: Is it a 2-approx?

↳ NO:



OPT: 6

Greedy: 8

$\frac{8}{6}$ maybe $\frac{4}{3}$-approx

Can blow this up & get worse...

$O(\log n)$-approx

Thm: Greedy VC is an $O(\log n)$ approximation:

$$\text{Greedy} \le \cancel{O(\log n) \cdot} \text{OPT}$$

$$k \cdot \log n \cdot \text{OPT}$$

for some constant k

pf: Let G_i = graph in i^{th} iteration, $|G_i|$ = # edges in G_i

Let d_i = max degree in G_i

```
GREEDYVERTEXCOVER(G):
  C ← ∅
  G_0 ← G
  i ← 0
  while G_i has at least one edge
      i ← i + 1
      v_i ← vertex in G_{i−1} with maximum degree
      d_i ← deg_{G_{i−1}}(v_i)
      G_i ← G_{i−1} \ v_i
      C ← C ∪ v_i
  return C
```

* let C^* be an optimal vertex cover of G

Note: $\sum_{v \in C^*} \deg_{G_{i-1}}(v) \ge |G_i|$

b/c C^* is a cover!

pf cont

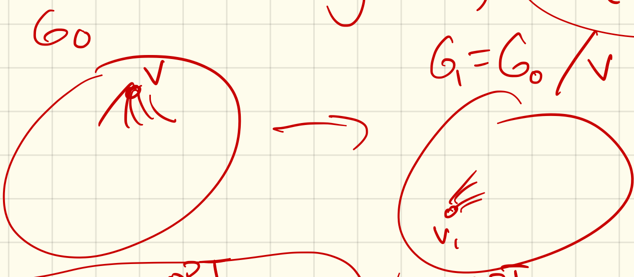
In other words, average degree in G_i is $\geq \frac{|G_{i-1}|}{OPT}$

Know d_i is max degree in G_{i-1}

so $d_i \geq \frac{|G_{i-1}|}{OPT}$

for $j > i$, $d_j \leq d_i$

$$\Rightarrow \forall j \geq i, \quad d_i \geq \frac{|G_{j-1}|}{OPT}$$



$$\Rightarrow \sum_{i=1}^{OPT} d_i \geq \sum_{i=1}^{OPT} \frac{|G_{i-1}|}{OPT} \geq \sum_{i=1}^{OPT} \frac{|G_{OPT}|}{OPT}$$

$$= |G_{OPT}| = |G| - \sum_{i=1}^{OPT} d_i$$

In other words:
first OPT iterations of loop
remove at least half
the edges of G.

So: after $OPT \lg |G|$
$\leq 2 OPT \lg n$ iterations,
all edges are gone.

In each round, choosing
one vertex

$\Rightarrow$ Greedy $\leq 2 OPT \lg n$

so $O(\log n)$-approx.

$\square$

A different approximation

Simpler idea:

- pick any edge + add its endpoints to the cover
- delete all "covered" edges
- Repeat

<u>DUMBVERTEXCOVER(G):</u>
$C \leftarrow \emptyset$
while G has at least one edge
    $(u, v) \leftarrow$ any edge in G
    $G \leftarrow G \setminus \{u, v\}$
    $C \leftarrow C \cup \{u, v\}$
return C

Runtime? $O(V+E)$

Thm DumbVertexCover is a 2-approximation.

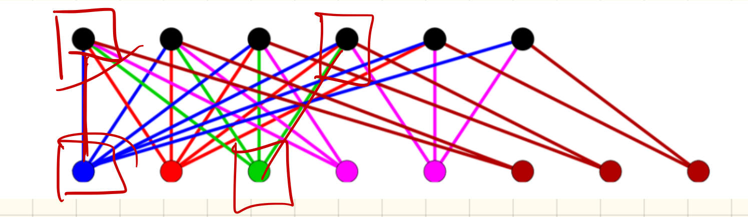
Pf: For each edge $e = \{u, v\}$, it must be covered, so either u or v is in C^*.

Worst case, my alg put 2 in (both u + v) instead of one.

$$\Rightarrow |C| \le 2|C^*|$$

$\blacksquare$

Huh?



Choosing both endpts breaks bipartite worst case.

Next time:

Traveling Salesman

Perhaps subset sum

(Most likely - problem day
on Friday)