Graph and Properties

Jonathan Irvin Gunawan

prerequisite

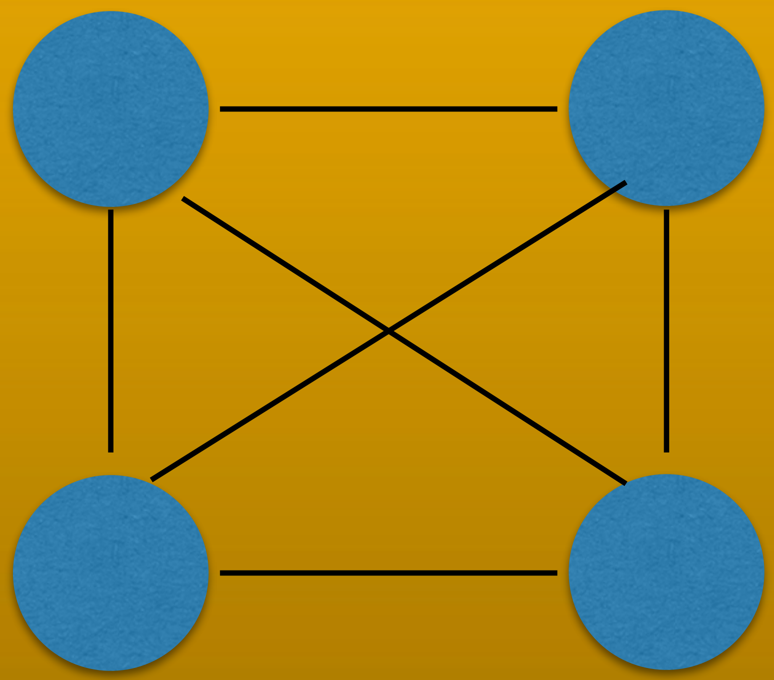
know graph

know maxflow

know proof by induction

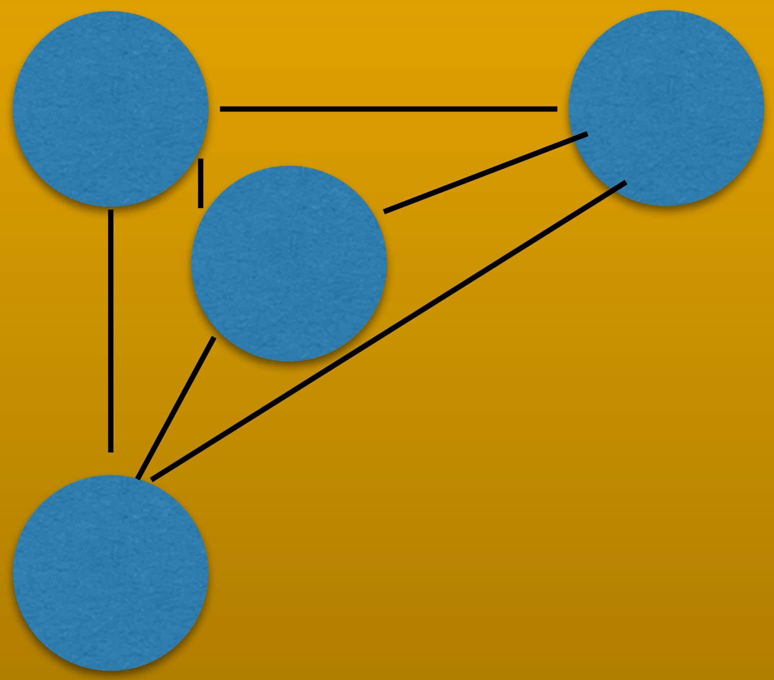
planar graph

a graph that **can be drawn** on a flat surface without crossing edge



planar graph

a graph that **can be drawn** on a flat surface without crossing edge



most important formula in planar graph: euler’s formula

$$V + F = E + 2$$

proof: induction

corollary

$E \leq 3V - 6$, for $V \geq 3$

proof: slightly complicated. on whiteboard instead

example

given unweighted graph, find APSP

$V \leq 1000$, $E \leq 10^6$

O(VE) no?

example

graph is planar

becomes O(V^2). cool

try to prove this

let's say there are n points and the distance between a pair of points $\geq$ 1cm. Prove that $\leq$ 3n pair of points with distance = 1cm connect two points with distance 1cm with an edge, then the graph is planar

more detail on whiteboard

$$E \leq 3V - 6$$

corollary 1 : there is a node with degree ≤ 5

proof by contradiction

$$E \leq 3V - 6$$

corollary 2 : six color theorem

know colouring graph?

proof by construction

choose any node with degree ≤ 5, let's say u.

remove that node, graph is still planar.
recursively colour the remaining graph.

since deg(u) ≤ 5, then there must be a colour that can be used for u that is not used by any of its neighbour

$$E \leq 3V - 6$$

theorem : five color theorem

proof by construction, similar with six, minor modification

$$E \leq 3V - 6$$

theorem : four color theorem

if we discuss the proof now, this camp ends next year theorem : four color theorem

maximum clique on planar graph?

if there is a clique of size X, then you will need X colours to colour the clique

so need $\geq X$ to colour the graph

thus max clique size ≤ 4

can check clique size 4,
check clique size 3,
check clique size 2,
...

$$1 \leq V \leq 3000$$
$$1 \leq E \leq 6000$$

bipartite graph

graph which vertices can be partitioned to L, R such that any edge connects (l, r) where l is in L and r is in R

bipartite graph

warming up

given a bipartite graph, find the L and R partition

easy

max bipartite matching

maxflow O(VE)

complicated, too long to code

alternating path algorithm

the idea is actually the same, but because the graph is bipartite, the code is much simpler

every time you are currently in R, you know that there is only one neighbour with residual flow > 0

```
bool match(int x) {
  for (int y : adj[x]) if (!visited[y]) {
    visited[y] = true
    if (with[y] == -1 || match(with[y])) {
      with[y] = x
      return true
    }
  }
  return false
}

int countMCBM() {
  int res = 0
  reset(with, -1)
  for (int i = 0; i < L; ++i) {
    reset(visited, 0)
    if (match(i)) ++res
  }
  return res
}
```

start with randomized pair

becomes fast

max weighted bipartite matching?

bipartite matching application

given a graph in the form of two complete disconnected subgraphs. find the max clique

easy

given a graph in the form of two complete ~~disconnected~~ subgraphs and many edges connecting these two subgraphs. find the max clique

not easy anymore

konig's theorem

max bipartite matching == min vertex cover

proof is omitted, but still feasible if you want to read

unlocks a whole new path

maximum independent set == V - min vertex cover

just take the nodes not in the vertex cover

this is for general graph

proof :

1 : if V’ is vertex cover, V - V’ is independent set

2 : if V’ is independent set, V - V’ is vertex cover

max clique == V - (min vertex cover in complement graph) proof :

1 : if V’ is vertex cover, V - V’ in complement graph is clique

2 : if V’ is clique, V - V’ in complement graph is vertex cover

go back to this problem

given a graph in the form of two complete ~~disconnected~~ subgraphs. find the max clique the graph is intentionally made so that the complement is bipartite find min vertex cover in the complement graph min path cover

given a directed graph, find the minimum number of path to cover all vertices

a path must not visit the same vertex twice cannot (not yet) be polynomial, because of hamiltonian path problem

min path cover

find the minimum number of path to cover all vertices **on DAG**

note that hamiltonian path on DAG is in P
how?

create L = V, R = V

for each edge (i,j) in V, add a new edge from L(i) to R(j) in your new graph

min path cover == |V| - max bipartite matching to understand why, we need to understand the representation of the matching

if L(i) is matched to R(j), we can say that the edge from L(i) to R(j) is used in one of our paths

number of path == number of nodes that has no outgoing edge used in the path == number of unmatched nodes in R what if the paths are not necessarily disjoint?

transitive closure

perfect matching

if size L == size R
the size of matching = size L is usually said to be a perfect matching

perfect matching

for each X subset of L, say
N(X) = {v : there is a u in X
such that v is a neighbour of u}

perfect matching

if there is a perfect matching, then for each subset X of L, $|N(X)| \geq |X|$

obvious, otherwise X does not have a perfect matching

perfect matching

the not so obvious one (and cool)

if for each subset X of L, $|N(X)| \geq |X|$, then there is a perfect matching

hall's theorem

perfect matching

proof : induction

if $|L| = 1$, obvious

perfect matching

assume theorem works for $|L|<m$, we prove that it works for $|L| = m$

case 1 : suppose for each proper subset S of L, $|N(S)| > |S|$

take any edge as a matching. the remaining graph satisfies the induction case

perfect matching

case 2 : suppose there is proper subset S of L, $|N(S)| = |S|$

graph $(S, N(S))$ satisfies induction case
graph $(L - S, R - N(S))$ satisfies induction case
we will have two disjoint perfect matchings

perfect matching

another sufficient condition

in regular bipartite graph, degree > 0, there is a perfect matching

not a necessary condition

dilworth’s theorem

in DAG, the minimum path cover == the maximum independent set

both u and v must not be in independent set if there is a path from u to v or v to u.

proof is also omitted but still feasible ok now some tasks

given a grid N*N, some of the cells are holes.

you want to put the minimum number of vertical/horizontal board of length N so that all holes are covered

$1 \leq N \leq 1000$

given a directed complete graph.
find the number of minimum path cover

```
int main() {
  puts(“1”)
}
```

given a bipartite graph, find edges which are part of ALL perfect matching

EOF

Q&A?