Lecture - 10 2/5/2015

Approximation Algorithms.

→ REVIEW

- Dynamic Programming
- Covering Times
- Basic Graph Theory

  Matching, Trees, MSTs, Hamilton / Euler circuits, Independent Sets, Cliques.

- Applications of Max Flow
- Dynamic Method
- The probabilistic method
- Reductions between {Vertex Cover,

DYNAMIC PROGRAMMING SOLN.

1. Find ~~a recursive Structure~~ Substructure (Find definition for problem) change problem a little.

   \# subproblems

   Time for subproblem

TSP

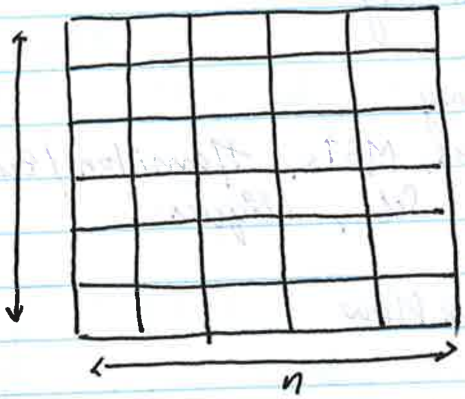
→ consider Hamiltonian paths. {look for end of the path, then end of the window}

→ Add ending.

2. Memoize: Memory: O (# of subproblems).

   Space ; memory × amount time used for subproblem.

→ Let's say



n decisions to go down
n decisions to go right.

for n by m grid $\binom{n+m}{n,m}$ ways to

→ Let's define Best(i,j) = max you can get starting at (i,j) and ending at (n,n)

What is the base case?

Base$(n,n) = 39$

$$= \max \begin{cases} a_{i,j+1} + \text{Best}(i,j+1) & (\text{if } j+1 \text{ is in bound}) \\ a_{i+1,j} + \text{Best}(i+1,j) & (\text{if } i+1 \text{ is in bounds}) \end{cases}$$

Final Answer:

$a_{11} + \text{Best}(1,1)$

→ Clever way of doing it.

from pg1.

3. Fill in array yourself without recursion. (Throw away parts of the memoization that aren't needed).

rec.

COVERING TIMES

1. $C(G) \leq 2m(n-1)$ → $R(G) = \max R_{uv}$

R_{uv} effective resistance.

2. $C(G) \leq 2me^3 R\ln(n) + n$
$= O(mR\log(n))$

→ ~~C(G)~~

Mathews Bound

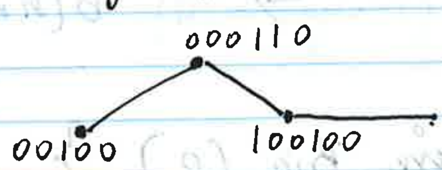
$n \leq C(G) \leq n\log(n)$

longest hitting time.

HYPERCUBE EXAMPLE:

→ Each node is a bit string of length h.

→ There is an edge if and only if two bitstring differ in exactly one location.

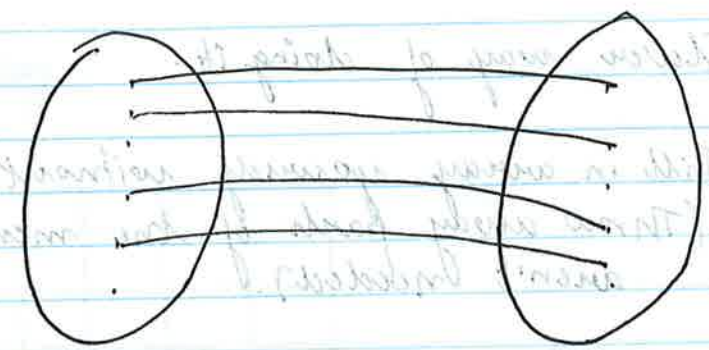


no. of nodes : $n = 2^h$

no. of edges $m = \frac{h \cdot 2^h}{2}$

degree of every node is h $\deg(i) = h$

$O(n^2\log(n))$



Even 1's Odd's.

$\exists$ Independent Set of size 2^{n-1}

Since bipartite only has 2 sides bigger than $\frac{n}{2}$.

→ Prove $C(Q_n) \leq O(nh^3)$

What is the diameter of Q_n = h {diameter is h at most}.

$R(Q_n) \leq h$

→ $C(G) \leq O(h \cdot n\} h \log(2^h))$
$= O(nh^3)$

Better $O(hn1.h) = O(nh^2)$

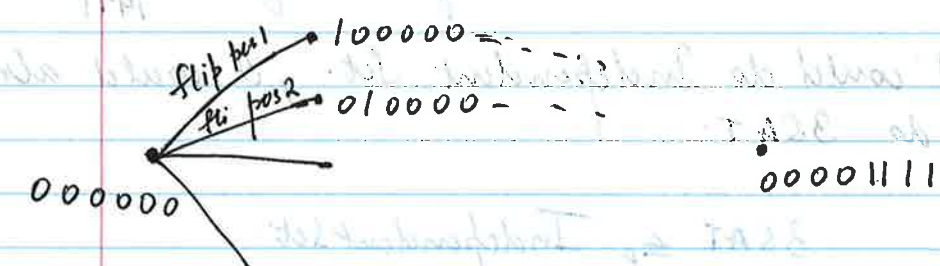
$\log(2^h)$ is order h.

{ $\log(2^h)$ is order h in Big (O) }

→ Show that max flow $Q_n \leq h$
{because max flow is bounded by lowest degree }.

→ Prove that max-flow is atleast h.
max-flow $\geq h$.

Show, there are



We have parallel nodes. Each path has resistance h.

$$R_{uv} \leq \frac{1}{\underbrace{\frac{1}{h}+\frac{1}{h}+\cdots\frac{1}{h}}_{h}} = 1$$

But bound in HW is more loose.
$(O(n^2 \log(n)))$ → not too tight.

2-SAT : Undirected chain ⇒ polynomial time algorithm.

? 3-SAT : directed chain ⇒ Exponential covering time

Worth thinking about Chain of Reductions?

$\leq_p$ SAT {SAT is the hardest problem of all by Cooke's Theorem} 1971

- If I could do Independent Set, I could also do 3SAT.

3SAT $\leq_p$ Independent Set.
↑ gadget construction
Relaxation for vertex cover.

Vertex Cover $\leq_p$ Integer Programming.
↖ Relaxation for Vertex Cover.

→ We also showed, if I could find, I could find vertex cover.

Independent Set $\leq_p$ Vertex Cover.
↑ complement.

SAT $\leq_p$ 3-SAT
↖ by De Morgan's Law.