Tuesday, February 27, 2024 8:52 AM

Backtracking

Recursion + try all possible sol'n.

(Reuse)

Prob 1: Maximum Independent set. (MIS)

Given undirect graph $n = |V|$

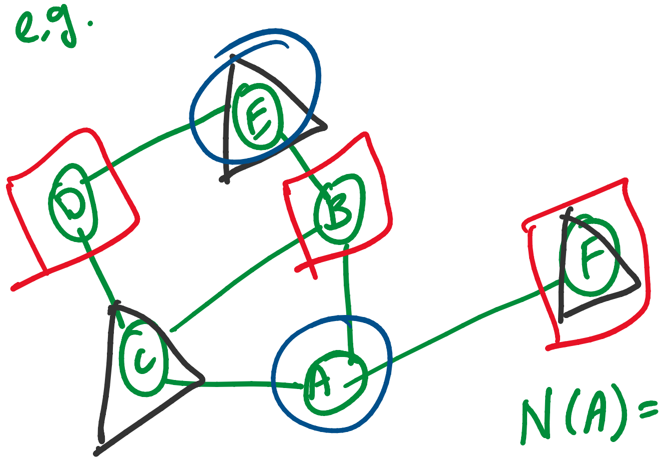
$m = |E|$

$G = (V, E)$.

optimization problems { Find $S \subseteq V$ maximizes $|S|$

s.t. $\forall u, v \in S$, $(u, v) \notin E$

e.g.



$S = \{B, D, F\}$ → opt. sol'n

$S = \{E, C, F\}$ → opt. sol'n

$S = \{A, E\}$

$N(A) = \{C, B, F\}$

Alg'm 0: Bruteforce

Try all possible subsets $S \subseteq V$

& check if S is I.S.

Time: $O(2^n \cdot m)$

Alg'm I: Backtracking.

idea: pick $v \in V$

case I: v is not in opt. sol'n

then remove v & all it's incident edges

$\downarrow$

recurse on $(G - v)$

case II: v is in the opt. sol'n

$S \leftarrow S \cup \{v\}$. & recurse on

$G - \underline{v} - \underbrace{N(v)}_{\text{friends of } v}$

where $N(v) = \{\underline{u} \mid (u,v) \in E\}$

MIS (G) {

// return the "size" of the max. ind. set.

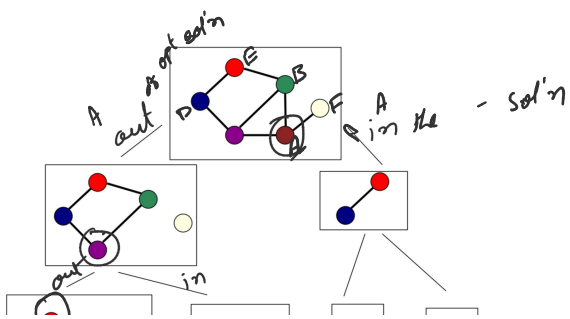
$O(1) \rightarrow$ 1. If $n=0$ return 0.

2. Pick vertex $\underline{v}$ in $\underline{G}$.

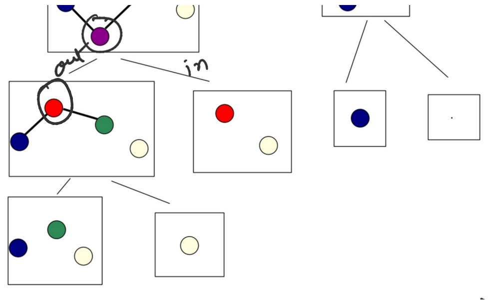
3. return $\max\{ \overbrace{MIS(G - v)}^{T(n-1)}$,

$1 + MIS(\underbrace{G - v - N(v)}_{T(n-1-deg(v))})\}$

}



Recursion will force bottom-up evaluation



evaluation

↓

Backtracking.

Time: $T(n) = T(n-1) + T(n-1-\deg(v)) + O(n)$

Analysis 0: $\deg(v) \geq 0$

$$T(n) \leq T(n-1) + T(n-1) + O(n)$$
$$= 2T(n-1) + O(n)$$
$$= 2\cdot(2T(n-2)) + 3O(n)$$
$$\vdots$$
$$= O(2^n) + O(2^n)\cdot O(n) = O(2^n \cdot n)$$

Analysis 1:

can include all isolated. deg-0 vertices in opt sol'n.

So just remove them from the graph. & put them in the sol'n. upfront.

$\deg(v) \geq 1 \Rightarrow T(n) \leq T(n-1) + T(n-1-1) + O(n)$

$$= T(n-1) + T(n-2) + O(n)$$
$$= O(\underline{x}^n \cdot n)$$

$$= O(x^n \cdot n)$$

[Fibonacci #s:

$$F_n = F_{n-1} + F_{n-2} \quad \text{if } n > 2$$
$$= \text{base case} \quad \text{if } n = 1, 2$$

guess $F_n \le x^n$

$$\left(x^n = x^{n-1} + x^{n-2}\right) \div \frac{1}{x^{n-2}}$$

$$\Rightarrow x^2 = x + 1$$

$$\Rightarrow \boxed{x = \frac{1+\sqrt{5}}{2}} \sim 1.618\ldots$$

golden ratio]

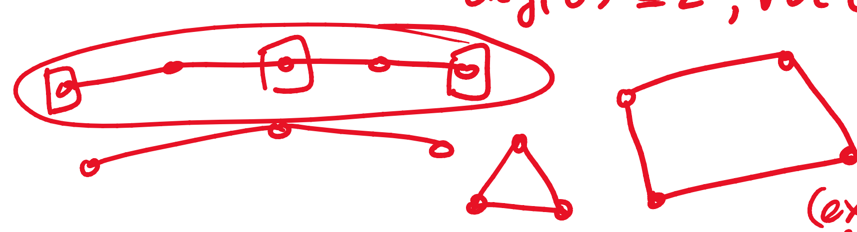
$$\boxed{O\left((1.618)^n \cdot n\right)}$$

Analysis:

idea: 2. Pick vertex v with maximum deg.

base case: if $\deg(v) \le 2$ then solve directly.

$\deg(v) \le 2, \forall v \in G$



(exe)

$\deg(v) \ge 3$.

$$\Rightarrow T(n) \le T(n-1) + T(n-4) + O(n)$$

$\Rightarrow T(n) \le T(n-1) + T(n-4) + O(n)$

$= O((1.381)^n \cdot n)$ $\quad \boxed{x^4 = x+1}$

Best known? $O(1.1991^n)$

[Xiao & Nagamochi '17]

Problem 2: Longest Increasing Subsequence

Given a sequence of #s.

$a_1, a_2, \dots, a_n$

Find a subseq $a_{i_1}, a_{i_2}, \dots, a_{i_\ell}$ maximizing ℓ

s.t. $i_1 < i_2 < \dots < i_\ell$

$a_{i_1} < a_{i_2} < \dots < a_{i_\ell}$

eg.: 4, 2, 3, 1, 10, 5, 17, 3, 9, 7, 12. ← opt. sol'n

Alg'm 0: Bruteforce

Try all subseq. & check if increasing

$\boxed{O(2^n \cdot n)}$

Alg'm 1: Backtracking

case I: a_n is not in opt. sol'n.

$\dots \langle a_1, \dots, a_{n-1} \rangle$

case I: a_n is not in opt

recurse on $\langle a_1, \dots, a_{n-1} \rangle$

case II: a_n is in opt. sol'n.

recurse on $\langle a_1, \dots, a_{n-1} \rangle$

but only consider ele $< a_n$.

value of last ele taken

$LIS(\langle a_1, \dots, a_n \rangle, X)$ {

// return "size" LIS s.t. every ele. in the subseq $< X$

$O(1) \rightarrow$ 1. if $n=0$ return 0.

$O(1) \rightarrow$ 2. if $a_n < X$

return max $\{ LIS(\langle a_1, \dots, a_{n-1} \rangle, X)$, $1 + LIS(\langle a_1, \dots, a_{n-1} \rangle, a_n) \}$

3. else return $LIS(\langle a_1, \dots, a_{n-1} \rangle, X)$

}

$LIS\text{-main}(\langle a_1, \dots, a_n \rangle)$

{ $LIS(\langle a_1, \dots, a_n \rangle, \infty)$ }

Analysis 0:
$$T(n) = 2T(n-1) + O(1)$$
$$= \boxed{O(2^n)}$$

Alternate View:

What is # of distinct subproblems the algo solves?

$$n \cdot (n+1) = n^2 + n$$

n: # prefixes of the seq. " Parameter 1

$(n+1)$: # possible values of x including ∞

We are solving the same problems over & over again!

idea: Store & Reuse. Memoization

$\boxed{a_{n+1} = \infty}$

$L[i,j]$ stores sol'n for i/p para. $(\underline{\langle a_1, \dots, a_i \rangle}, X = a_j)$

$L[i,j] =$ un-det

$a_{n+1} = \infty$

... memoization:

$L[i,j] = $ un-def

$a_{n+1} = \infty$

Alg'm 2 w/ memorization:

LIS(i, j) {

// returns size of LIS in $\langle a_1, \dots, a_i \rangle$ where max ele $< a_j$

1. if $i = 0$ then return $L[i,j] = 0$
2. if $L[i,j] \neq$ un-def return $L[i,j]$
3. if $a_i < a_j$ then

$$\text{return } L[i,j] = \max \begin{Bmatrix} \text{LIS}(i-1, j), \\ 1 + \text{LIS}(i-1, i) \end{Bmatrix}$$

4. Else return $L[i,j] = \text{LIS}(i-1, j)$

}