s'

$\frac{2}{3}n$

LAST TIME.

- Analysis

DIVIDE & conquer algo →

- string search →

- RANDOMIZED Algorithms ⇒

$T(n) = T(\text{subset}) + \boxed{\text{work at level}}$

Closest pair algo -

Problems & Class they belong to based on an ability to solve them & How Quickly.

Class P: A problem A belongs to P iff $\exists$ an algorithm. that solves A in $O(n^k)$ $k = \text{constant}$.

$= O(n^{O(1)})$

B: Problem: Find all occurances ~~searching~~ a reverse pattern P in text T.

$B \in P\text{-class}$?

Yes proof by providing an algorithm that is poly-time.

Does deterministic selection algo belong to class P? NO

C Problem: Find the kth smallest item in a list of n unsorted items.

$C \in P$ proof:

$\exists$ also deterministic selection where det. selection has a asymp. run-time of $O(n)$. Where n is size of input.

quicksort $\in P$? NO

given: a set $S = S_1, S_2, S_3 \ldots S_n$

output: a permutation of S such that $S_1 \leq S_2 \leq S_3 \ldots \leq S_n$

FIND X

given: a item X ~~and sta.~~ and a set S st. S contains n items sorted.
and the time to compare $X \in S[i]$ is $O(n)$.
Find if $X \in S$

$X \in P$?

Yes, proof: by construction

~~use binary~~ search algo but overload the comparison opt. to handle objects called items.

```
searchitem( item X, S)   lo=0; hi=|S|-1
    m = (lo+hi)/2
    if (X ≤ S[m])        comparison of items
        let hi = m-1
while(lo<high)
    else
        lo = m
return S[lo] == X;
```

$(\text{Log}\, n) \times n = O(n^2)$

problem show in $\in$ P.

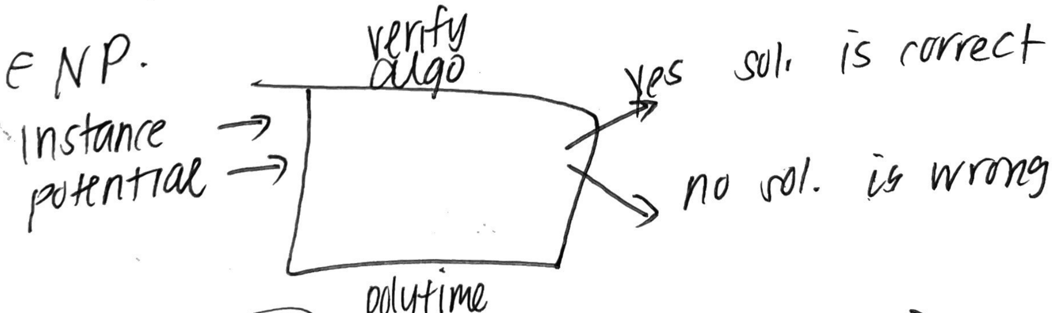
prove it by giving either known algo
w/ known runtime
or creating one.

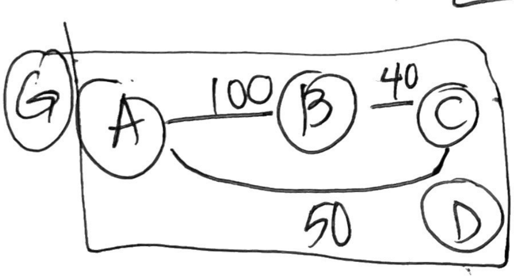
NP : none deterministically polynomial

problem is in class of NP iff exist a verifier (algo) that is polytime $O(n^k)$ that given the instance of the problem and a potential solution, verifies the solution for that problem.

SP · problem: shortest path on a graph btw two ~~point~~ ~~vertex~~ nodes.

SP $\in$ NP.

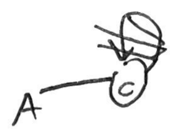




problem(G, (~~A~~, C))

solution: A → B → A → C
100 + 100 + 50 = 250

~~A→~~ ~~B~~ A → D → C



bool verify (G, (startN, endN), path P)

$n^2 \rightarrow$ int D = call Dijstra(G, startN) // shortest paths from startN.

int weight = 0

$O(n) \rightarrow$ for (~~xi xi+1 ∈ path~~, i=1 to |path|)

weight = weight + w(~~xi, xi+1~~)

where $(x_i, x_{i+1}) \in$ path

$O(1)$ if (weight != D[endN])

return false

$O(1)$ else

$\boxed{\text{return true}}$

verify = $O(n^2)$ Hence $\exists$ algo that verifies shortest path problem in $O(n^2)$ i.e. poly

$\Rightarrow$ shortest path $\in$ NP.

Shortest path $\in$ P yes proof:

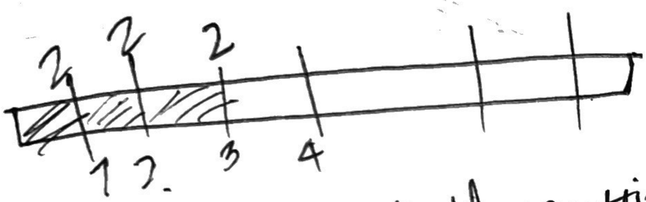
$\exists$ an algo Dijkstra st. Dijstra is $O(n^2)$

AND Hence SP $\in$ P.

2 Classes

① P poly class is a set of ~~Algo~~ problem that have algorithms that solve them in polytime.

② NP class is a set of problems that verified (double check a potential sol.) in poly-time, given the sol @ problem.



n Rod-cutting $\in P$

solution to Rod cutting

list all possible cut stategies $O(2^n)$

AND check is they are bring $O(n)$

max revenue

$2^n \cdot n$.

Yes NOT B.C. of above but ECS122a.

Rod-cutting is $O(n^2)$ ~~a~~ using dyn. prog.

k-IS $\in$ NP b.c. $\exists$ verify IS $= O(n^3)$.

verifyIS(G, k, I, k) ::

if ($|I| < k$) return false

// $I = \{x_1, x_2, \dots x_{k'}\}$ $\quad k' = |I|$

$O(k')$ for ($i=1$ to k') {

$O(k)$ for ($j=i+1$ to k') {

$O(n)$ - if ($(x_i, x_j) \in E$) $O(n = |V|)$

return false

else continue

}

}

$k' \cdot k' \cdot n$
$= (k')^2 n$

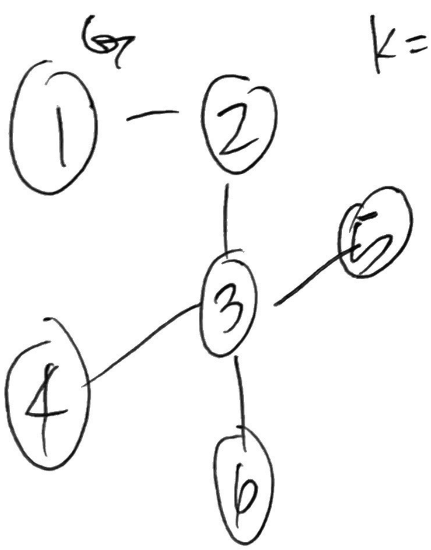
n^3 for Adj List

n^2 for Adj Matrix

return true;

$K = O(|V|)$

$k \geq n$



$k = 4$ $\quad I = \{1, 4, 6, 5\}$

$(1, 4) \in E$ NO

$(1, 6) \in E$ NO

$(1, 5) \in E$ NO

$(4, 6) \in E$ NO

$(4, 5) \in E$ NO

$(5, 6) \in E$ NO

NP > k-IS → verify in polytime