Complexity	Theory
------------	--------

Defu: A Turing Machine	includes the following
	set of states Q= {80,, 8m}
	m mal
2 & "alphohat"	- a Set of Symbols
A. 2 11 1001	or 201 0 - Symbolis
3. f "transition fo	netion" f: QXZ QXZ U {e}
	empty symbol
4. Tape: Input,	
Input tape 0 1 0	
Output tope	
Storage trae	
Example: Suppose $\xi =$	{o, 1, ₩}
(g_0) (g_1)	0/1
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	
(43) E (4))0/1
$(q_3) \leftarrow (q_2)$	
Uils.	}
Input tape 0010	1 # 1 0 #
Storage tape	
output take	#15 = # 0's before first #
Any reasonable model of con	uputation is equivolent to
a turing	

Time complexity N= Size of input T(n) = # of steps/transitions the tuling muchine makes Space complexity S(n) = amount of storage required Example: important to consider your imput!!! Adding: atb "unwy encoding" 111111 # 11111 Tich = GA 111111 11111 length atb. atb Versus: binery encoding 101 # 011 a base 2 b base 2 T2 (n) = C2 N 1110 only log(a) + log(b) length ath base 2 * both linear time algorithms, but Ta is exponentially better than Ti

Time complexity N= Size of input T(n) = # of steps/transitions the turing muchine makes Space complexity S(n) = amount of storage required Example: important to consider your imput!!! Adding: a+6 "unwy encoding" 11111 # 11111 Tich = GA length atb. atb Versus: binery encoding #1011 101 a base 2 b base 2 Ta(n) = Can 1110 only log(a) + log(b) length ath base 2 * both linear time algorithms, but Ta is exponentially better than Ti

	Defn: A problem is a set of strings on E
	An instance is a particular String from the publican
	Defor: The class of polynomial time problems P is
	the set of problems for which there exists a
	turing machine with polynomial time comprexity.
	Reduction: Problem B reduces to problem A if for
	all instances of B there exists a map
1	R:B > A , R(b) EA , S.t. Ris polytime.
1	
1	Detn: A and B are polynomially equivalent if A reclues to B
1	and B reduces to A.
	Example: Vertex Cover
	· Optimization asks for the minimum vertex cover
	· Feesibity asks for a vertex cover of size & k
	S'table Set
	· optimization asks for the maximum Stable Set
	of leasibility asks for stateset of size 2 n-K
	All of these are polynomially equivalent
-	Defn: A non-deterministic polynomial time (NP)
	turng machine has a map
	f:QX E → 2QX E U Ee3
	it can comport parallel computations.
-	Alternative Deto: A problem is NP if there exists a polynomial
	time algorithm to check a given certificate.

Clearly PENP. Probably PINP.

	Defn: A problem is a set of strings on E
	An instance is a particular String from the problem
	Defu: The class of polynomial time problems P is
	the set of priblems for which there exists a
	turing machine with polynomial time comprexity.
_	
	Reduction: Problem B reduces to problem A if for
	all instances of B there exists a map
	R:B > A , R(b) EA , S.t. Ris polytine.
	Detn: A and B are polynomially equivalent if A recluces to B
	and B reduces to A.
	Example: Vertex Cover
	· Optimization asks for the minimum vertex cover
	· Feasibity asks for a vertex cover of size & K
	S'table Set
-	· optimization asks for the maximum stable set
	· feasibility asks for stateset of size Zn-K
1	All of these are polynomially equivalent
-	
-	Detn: A non-determinate polynomial time (NP)
	tunne machine has a map
-	f:QXE > 2QXEU Ee3
	it can comport parallel computations.
	Atternative Defor: A problem is NP if there exists a polynomial
	time algorithm to check a given certificate.
	J

Clearly PENP. Probably P+NP.