Polynomial Time Complexity

Theory of Computer Algorithms
Jake Clements
April 21, 2005

Adding Problem

```
Add(Integer n) {
   Integer sum=0;
   for i=1 to n DO {
        sum=sum+1;
   }
}
```

O(n) = Linear = Polynomial of degree 1

Halting Problem

```
Halt (Program A, input x) {
   If A(x) halts {
        print("Good program");
   } else {
        print("Bad program");
   }
}
```

Adding Problem Revised

```
Add(Integer n) {
   Integer sum=0;
   while(n<0) {}
   for i=1 to n DO {
        sum=sum+1;
   }
}
```

Definitions

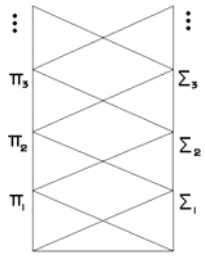
- A problem must have a deterministic algorithm that *decides* it
- The algorithm must be able to be represented by some polynomial expression in terms of the problem size
- The algorithm has *polynomial time complexity*
- The problem is *polynomial time complete*

Arithmetic Hierarchy

The set where π_1 intersects with Σ_1 (known as π_0 or Σ_0) is the set of all languages for which a decider can be constructed.

A *decider* is a Turing Machine that halts on all inputs.

In other words, this is the set of all languages for which a program can be written that halts on all inputs.



Complexity Classes

- P – the class of decision problems that can be solved on a deterministic sequential machine in polynomial time
- NP - the class of decision problems that can be solved in polynomial time on a non-deterministic Turing machine

Polynomial vs. Exponential

	1	10	100	1000
n^2	1	1E2	1E4	1E6
n^3	1	1E3	1E6	1E9
n^5	1	1E5	1E10	1E15
n^{10}	1	1E10	1E20	1E30
2^n	2	1E3	1E30	1E301

References

- Sipser, Michael. Introduction to the Theory of Computation. PWS Publishing Company, 1997.
- http://en.wikipedia.org/wiki/Polynomial_time

Polynomial Time Algorithms

Ray Wallace

Classes of Time Complexity

- Polynomial Time (P)
- Non-Polynomial Time (NP)
- NP-Complete (NPC)

What's it Mean?

The class of Polynomial-Time algorithms is "all those decision problems that can be solved on a deterministic sequential machine in an amount of time that is polynomial in the size of the input"*

* Wikipedia, entry for "Complexity classes P and NP"

Example Complexities

Polynomial Time:

- $O(n)$
- $O(n^2 + n)$
- $O(10^{100}n)$

Non-Polynomial Time:

$O(e^{2n})$
$O(2^n)$
$O(e^{n/1000})$

Equivalencies

- log(n) space complexity: L
  - $L \subset P$
- Polynomial space complexity: PSPACE
  - $P \subset PSPACE$
- Non-Polynomial time complexity: NP
  - ????

Oracle Machines

- Use NP problems in P algorithms
  - Normal Turing machine plus an "oracle"
- Notation:

P algorithm using an NP oracle:
P^{NP}

P algorithm using a Satisfiability oracle:
P^{SAT}

References

Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald Rivest

www.csc.liv.ac.uk/~ped/teachadmin/algor/npcomp.html

www.wikipedia.org entries for Complexity classes P and NP, Computational complexity theory, Oracle machine, etc.