COL 702   Lecture 1 July 23

Time and Space complexity.

Upper bounds

: achievable $O(f(n))$

Lower bounds : $\Omega(g(n))$

No algorithm can be better

$f(n) \geq g(n)$

$f(n)$ and $g(n)$ are "asymptotically" in the same class

Primitive "instructions"

Set of instructions { Read, Write, Arithmetic, comparison, Logical }

"Tightness of the analysis using a concrete input" shouldn't be confused with lower bound

DO NOT WRITE CODE

```
a := 2
for i := 1 to n do
    a ← a * a
end
print a
```

what is the value printed 2^{2^n}

Notion of induction

Claim After i iterations, $i \geq 0$, value of $a = 2^{2^i}$

Base case : $i = 0$ $a = 2^{2^0} = 2^1 = 2$

Inductive step : If assertion $P(i)$ is true after i steps then it is true after $i+1$ steps $P(i) \Rightarrow P(i+1)$ for all $i \geq 0$

Size of operands is crucial for analysis : Operands must fit into a few "words"

For inputs of size n, the operands should be $O(\log n)$ bits

Given a set S of n pairs of the form (x_i, y_i) $1 \le i \le n$

A pair (x_i, y_i) "dominates" (x_j, y_j) iff $x_i \geq x_j$ and $y_i \geq y_j$

A maximal subset of S is ~~are~~ those pairs that are not dominated.

<u>Problem</u> : Find all maximal pairs