CPSC 331

T#5

Worse Case Running Time

Public static Boolean distinctEntires (int[] A){	**cost**	**time**
for (int i=1;i<A.lenght;i++) {	c1	n
for (int j = 0; j<i; j++){	c2	~~(n-1)(n)~~ 2+3+…+n
if (A[j] == A[i]) {	c3	~~(n-1)(n-1)~~ 1+2+3+…+(n-1)
return false;	c4	1
};		
};		
};		
return true;	c4	1
}		

- In order to ease our analysis of the procedure:
  - Ignore the actual cost of each statement, c_i

  These constants give us more detail than we really need

  - We need one more simplification

  the order of growth of the running time really interests us . So consider only the leading term of a formula

  Since the lower-order terms are relatively insignificant for large values of n.

  - Ignore the leading term's constant coefficient

Growth of Functions

- For a given function g(n) we have :
  - Θ(g(n)) = {f(n) : there exist positive constants c_1, c_2 and n_0 such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)\ for\ all\ n \ge n_0\}$
  - Function f(n) belongs to the set Θ(g(n))

  - O(g(n)) = {f(n) : there exist positive constants c and n_0 such that $0 \le f(n) \le c\quad g(n)\ for\ all\ n \ge n_0\}$
  - To give an upper bound of a function

  - Ω(g(n)) = {f(n) : there exist positive constants c and n_0 such that $0 \le c\quad g(n) \le f(n)\quad for\ all\ n \ge n_0\}$

Relations Between Θ, *O*, Ω