Devin Hardy

CS 472

1.

O(n) is O(c^n) because it enumerates r by checking the pairing of i and j and the subset of k

2.

O(n^3): n^2 + 3n^3 the largest exponent or n is 3 so 3n^3 >= cn^3 if c = 1 1n^3 = n^3

Ω(n^3): n^2 + 3n^3 the largest exponent for n is 3 so 3n^3 <= cn^3 if c = 1 1n^3 = n^3

3.

2^(n+1) -> Θ(2^n):

For 2^(n+1) -> O(2^n) to exist 2^(n+1) = 2*2^n so that 2*2^n <= c*2^n for any c >= 2

For 2^(n+1) -> Ω(2^n), Ω(n) requires there exist a constant c > 0 s.t., f(n) >= c*g(n). This is satisfied for any constant 0 < c <= 2. As 2^(n+1) -> O(2^n) and 2^(n+1) -> Ω(2^n), then 2^(n+1) -> Θ(2^n).

A^n+1 -> Θ(A^n):

A^n+1 -> O(A^n)? f(n) -> O(g(n)) iff ∃c s.t. for, f(n) <= c* g(n). With the definition of exponents, A^(n+1) = A*A^n so A*A^n <= C*A^n for any C >= A;

A^n+1 -> Ω(A^n)? Ω(n) requires there exist a constant c > 0 s.t., f(n) >= c*g(n). If A = 0, 0^n would still be 0 so A > 0. As A^n+1 -> O(A^n) and A^n+1 -> Ω(A^n), then A^n+1 -> Θ(A^n).

4.

Worst case the order is n^2; The function has to check if the matrix is complete which requires checking every pair of the matrix.

5. Algorithm for Gray Code to solve Tower of Hanoi.

2^n – 1 algorithm for Tower of Hanoi for nth disks [1]

6.

Cited

[1] A. Mishra, “Tower of Hanoi Recursion Game Algorithm explained,” *HackerEarth Blog*, 13-Sep-2021. [Online]. Available: https://www.hackerearth.com/blog/developers/tower-hanoi-recursion-game-algorithm-explained/. [Accessed: 31-Jan-2022].