Algorithm: Homework #1

Due Date: April 18, 2025

1. Give the pseudo-code of a logarithmic-time ($\theta(\log n)$-time) algorithm for computing the n-th Fibonacci number.

θ(logn) 인 의사코드 찾기

2. Prove that $n^2 \in o(2^n)$ using the formal definition of small-o notation.

엄밀한 증명

3. Give the closed forms for the following recurrence relations. Solve them using their characteristic equations.

(3a) $f_{n+2} = f_{n+1} + f_n \, (n \geq 0)$, $f_0 = 0, f_1 = 1$

(3b) $f_{n+2} = f_{n+1} + f_n + 1 \, (n \geq 0)$, $f_0 = 0, f_1 = 1$

(3c) $T(n) = 3T(n/3) + 2n/3$, $T(1) = 0$ (You may assume that $n = 3^k, k \geq 0$)

(3d) $T(n) = 3T(n/3) + 2n$, $T(1) = 0$ (You may assume that $n = 3^k, k \geq 0$)

4. Consider **MergeSort** algorithm in the textbook.

(4a) Given array size n, find a recurrence relation for the ***best-case*** time complexity for **MergeSort**. best-case 점화식 찾고

(4b) Solve the recurrence relation for (1), given n ($= 2^k$ for some integer $k > 0$).

풀이

5. (**Theoretically Fast QuickSort**) There exists a linear-time ($O(n)$-time) algorithm that computes a median value among given n values. Assume that we have already known this algorithm. Using this algorithm, **(5a)** design a variant QuickSort algorithm of which the worse-case time complexity is $O(n \log n)$ (just give its pseudo-code) and **(5b)** prove that its time complexity is $O(n \log n)$.

Final location of a pivot:

$\left\lfloor \frac{low + high}{2} \right\rfloor$-th value

선형시간에
중앙값 구하는거 써서
항상 반씩 나눈다

그에 대해 최악시간복잡도 분석하기