CSE 101 Homework 1

Kreshiv Chawla, Brian Masse, Taira Sakamoto, Emily Xie

October 7, 2025

1. (a)
$$\frac{2^{n/2}}{2^n} = 2^{-n/2} \xrightarrow[n\to\infty]{} 0.$$
Hence, for any constant $c > 0$ and all sufficiently large n, $2^{n/2} < c\,2^n$. **False.**

(b)
$$(n^3+2n+1)^4 = n^{12}\left(1+2n^{-2}+n^{-3}\right)^4 = n^{12}\left(1+O(n^{-2})\right),$$
$$(n^4+4n^2)^3 = n^{12}\left(1+4n^{-2}\right)^3 = n^{12}\left(1+O(n^{-2})\right).$$
Therefore,
$$\frac{(n^3+2n+1)^4}{(n^4+4n^2)^3} \xrightarrow[n\to\infty]{} 1 \neq 0.$$
Not little-o. **False.**

(c)
$$\log\left(n^{10}\right) - \log(\log n) = 10\log n - \log\log n.$$
Hence $\Theta(\log n)$. **True.**

(d)
$$\sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2 = \frac{n^2(n+1)^2}{4} = \frac{n^4+2n^3+n^2}{4}.$$
True.

(e)
$$n! = (n)(n-1)(n-2)\cdots 2\cdot 1 \le n\cdot n\cdot n\cdots n = n^n.$$
True.

2. **Base case.** For $n = 2$,
$$F_2 = F_{2-1} + F_{2-2} = F_1 + F_0 = 1 + 0 = 1.$$

Inductive hypothesis. Assume that the statement about F_n holds for all integers up to k.

Inductive step.
$$F_{k+1} = F_k + F_{k-1} = (F_{k-1} + F_{k-2}) + F_{k-1} = F_k + F_{k-1},$$
establishing the step.

3. **Base case.** Initially, there are 101 black marbles (odd).

Inductive hypothesis. Assume that after k moves, the number of black marbles has the same parity as it had initially (odd).

Inductive step. On move $k{+}1$, one of three things happens:

- BB: black count changes by -2 (parity unchanged).

- RR: black count changes by 0 (parity unchanged).
- BR: black count changes by 0 (parity unchanged).

In every case, the parity of the black count after move $k+1$ equals the parity after move k. By the inductive hypothesis, this equals the initial parity. Hence, the claim holds for $k+1$.

4. Following the Algorithm Defined in HW1. Consider the case when $A[i]+A[j] = V$ for some $1 \le i < j \le n$. Prove the invariant: If the while loop has not terminated, then $I \le i < j \le J$.

   (a) **Base case:** $I = 1, J = n$ (assuming $n > 1$).

   $$1 \le i < j \le n \implies I \le i < j \le J.$$

   (b) **General case:** Assume that after $x > 1$ iterations, $I \le i < j \le J$. Show the loop invariant after the $x+1$-th iteration.

   Case 1: $i = I$ and $j = J$

   $$A[i] + A[j] = V \implies \text{returns True.}$$

   Case 2: $j = J$

   $$\begin{aligned} &I < i, && (1)\\ &\text{By sorted array: } A[I] + A[J] = A[I] + A[j] < V, && (2)\\ &I{+}{+}, && (3)\\ &I \le i < j \le J. && (4) \end{aligned}$$

   Case 3: $i = I$

   $$\begin{aligned} &J > j, && (5)\\ &\text{By sorted array: } A[I] + A[J] = A[i] + A[J] > V, && (6)\\ &J\text{–}, && (7)\\ &I \le i < j \le J. && (8) \end{aligned}$$

   Case 4: Else

   $$\begin{aligned} &J > j \text{ and } i < I, && (9)\\ &\text{Either } I++ \text{ or } J-- \text{ will uphold the loop invariant.} && (10) \end{aligned}$$

5. *Algorithm:*

   (a) Initialize NextLarger[1..n] $\leftarrow n+1$ and an empty stack S.

   (b) For $k = 1$ to n:

i. While S is not empty and $A[S.\text{top}] < A[k]$:

$$i \leftarrow S.\text{pop}(); \quad \text{NextLarger}[i] \leftarrow k.$$

ii. Push k onto S.

Indices left in S have no larger element to their right, so their value remains $n+1$.

Proof: Define invariant $I(k)$: after processing elements $A[1..k]$,

(a) The stack S contains indices of a strictly decreasing sequence of A-values.

(b) For every $i \leq k$ not in S, NextLarger$[i]$ is correctly set.

(c) For every i in S, $A[i] > A[t]$ for all $i < t \leq k$.

Base case: $k = 1$ satisfies $I(1)$.

Inductive hypothesis: Assume $I(k-1)$ holds after processing the first $k-1$ elements.

Inductive step: When processing $A[k]$, every i popped from S satisfies $A[i] < A[k]$ and, by (c), there was no t with $i < t < k$ and $A[t] > A[i]$. Thus k is the smallest index $> i$ with $A[k] > A[i]$, so NextLarger$[i] = k$ is correct. After popping, S remains strictly decreasing, and pushing k preserves (a)–(c). Hence $I(k)$ holds. By induction, $I(n)$ holds, so all NextLarger$[i]$ are correct.

Time complexity: Each index is pushed and popped at most once. All other operations are $O(1)$. Hence, the total time is $O(n)$ and the space is $O(n)$.