1

Question A

Mark for each of the following statements if they are true or false. In each case prove your answer. You can use the following known inequality without proving it: $\log n < n$ for $n > 1$.

1. $n^2 + 2n\log n + 5 \in O(n^3)$
2. $8n^3 + 2n + 20\log(n^{12}) \in \Theta(n^3)$
3. $2^n \in \Omega(9^{\frac{n}{3}})$

1. $n^2 + 2n\log n + 5 \in O(n^3)$ True

we know that for any $n>1$, $\log n<n$. Therefore, we get $n^2+2n\log n+5<n^2+2n^2+5$.

we'll prove by induction over $n\in\mathbb{N}$ that for every $c=1$, $n_0\geq 4$ we get $n^2+2n^2+5=3n^2+5\leq n^3$.

- Base case ($n=4$): We get $3\cdot 4^2+5=3\cdot 16+5=53\leq 4^3=64$. Thus, the claim holds.
- Induction hypothsis: Assume the claim holds for $k\geq 4$, $k\in\mathbb{N}$, $n=k$. That is, $3k^2+5\leq k^3$.
- Induction step: We'll prove the claim hold for $k+1$.

  $3(k+1)^2+5=3k^2+6k+3+5$. By the induction hypothsis we get that $3k^2+5\leq k^3$. Thus, $3k^2+6k+3+5\leq k^3+6k+3$.

  When looking on the other side of the equation we get $(k+1)^3=k^3+3k^2+3k+1$. Since $k\geq 4$, we get that $k^3+6k+3\leq k^3+3k^2+3k+1$. Thus we get $3(k+1)^2+5\leq(k+1)^3$.

  That is, the claim holds for $k+1$.

Therefore, by the induction, we deduce that for every $n\geq 4$, $c=1$ it holds that $3n^2+5\leq n^3$. Thus $n^2+2n\log n+5\leq n^3$.

That is, $n^2+2n\log n+5\in O(n^3)$.

2

2. $8n^3 + 2n + 20\log(n^{12}) \in \Theta(n^3)$ True

By logarithmic identities we get that
$8n^3+2n+20\log(n^{12}) = 8n^3+2n+240\log n$.

First, we'll prove that $8n^3+2n+20\log(n^{12}) \in \Omega(n^3)$.

We'll prove by induction that for every $n \ge 8$, $c=1$
$8n^3+2n+240\log n \ge n^3$.

Base case ($n=8$): $8\cdot 8^3+2\cdot 8+240\log 8 = 4832 \ge 512$.

Induction hypothsis: Assume that for $k \ge 8, k \in \mathbb{N}$, $k=n$ the claim holds. That is $8\cdot k^3+2k+240\log k \ge k^3$

Induction step: We'll prove the claim hold for $k+1$.

$8\cdot(k+1)^3+2(k+1)+240\log(k+1) = 8(k^3+3k^2+3k+1)+2k+2+240\cdot\log(k+1) =$
$8k^3+24k^2+24k+8+2k+2+240\cdot\log(k+1)$

By the Assumption $8k^3+2k+240\log k \ge k^3$. Therefore, since log is decreasing function, we deduce that
$8k^3+2k+\log(k+1) \ge k^3$. Thus $k^3+24k^2+24k+10 > k^3$
when looking on the other side of the equation we get
$(k+1)^3 = k^3+3k^2+3k+1$. Thus since $k \ge 8$ we get that
$k^3+24k^2+24k+10 > k^3+3k^2+3k+1$. That is,
$8\cdot(k+1)^3+2(k+1)+240\log(k+1) > (k+1)^3$. Hence, the claim holds for $k+1$.

Therefore, by induction we get that $8n^3+2n+240\log n \ge n^3$ for any $n \ge 8$, $c=1$.

That is $8n^3+2n+20\log(n^{12}) \in \Omega(n^3)$.

Secoundly, we'll show that $8n^3+2n+20\log(n^{12}) \in O(n^3)$.

Since for any $n > 1$ $n > \log n$, we get that
$8n^3+2n+240n > 8n^3+2n+240\log n$.

Thus, we'll prove be induction over n that for any $n \ge 8$, $c=12$.

3

$8n^3+2n+240n \le 12n^3$.

Base case ($n=8$): $8\cdot 8^3+2\cdot 8+240n = 6032 \le 12\cdot 8^3 = 6144$

Induction hypothsis: Assume that for $k\ge 8$, $k\in\mathbb{N}$, $k=n$ the claim holds. That is $8\cdot k^3+2k+240k \le 12k^3$.

Induction step: We'll prove the claim hold for $k+1$.

$8(k+1)^3+2(k+1)+240(k+1) = 8k^3+24k^2+24k+8+2k+2+240k+240$. By the assumption $12k^3 \ge 8k^3+2k+240k$, thus we get that

$8k^3+24k^2+24k+8+2k+2+240k+240 \le 12k^3+24k^2+24k+8+2+240$.

When looking on the other side of the equation we get $12(k+1)^3 = 12k^3+36k^2+36k+12$. Thus since $k\ge 8$ we get $12k^3+24k^2+24k+250 \le 12k^3+36k^2+36k+12$. Hence, the claim holds for $k+1$.

That is, by the induction we get that $8n^3+2n+240n \le 12n^3$.

Therefore, $8n^3+2n+20\log(n^{12}) \le 12n^3$ for any $n\ge 8$, $c=12$. That is $8n^3+2n+20\log(n^{12}) \in O(n^3)$.

Thus, we get $8n^3+2n+20\log(n^{12}) \in \Theta(n^3)$.

3. $2^n \in \Omega(9^{\frac{n}{3}})$ False

Assume toward contradiction that $2^n \in \Omega(9^{\frac{n}{3}})$. That is, There exist $n_0, c>0$ s.t for any $n\ge n_0$ $2^n \ge c\cdot 9^{\frac{n}{3}}$.

$2^n \ge c\cdot 9^{\frac{n}{3}} \Rightarrow 2^{3n} \ge c^3\cdot 9^n \Rightarrow \frac{2^{3n}}{9^n} \ge c^3 \Rightarrow (\frac{2}{9})^{2n} \ge c^3 \Rightarrow \log_{\frac{2}{9}} c^3 \ge 2n \Rightarrow$

$\frac{\log_{\frac{2}{9}} c^3}{2} \ge n$. We get contradiction, since n is bounded by

$\frac{\log_{\frac{2}{9}} c^3}{2}$. Therefore $2^n \notin \Omega(9^{\frac{n}{3}})$.

Question B

For each of the following proofs state whether it is correct and explain where the proof fails.

1. **Claim:** $8n^2 \notin O(n^2)$.
   Proof: For every $n \geq 1$ it holds that $8n^2 > n^2$. Hence using $n_0 = 1, c = 1$ we conclude that $8n^2 \notin O(n^2)$.

2. **Claim:** $n \in \Omega(n^3)$.
   Proof: We will prove by induction on n, that there exists a constant $c > 0$ such that for every $n \geq 2$ it holds that $n \geq c \cdot n^3$. This is enough in order to show $n \in \Omega(n^3)$.
   Base: for $n = 2$, for $c = 1/4$ it holds that $2 = \frac{1}{4} \cdot 8 = c \cdot 2^3$.
   Assumption: Assume that the claim is correct for $n-1$, meaning there exists $c_1 > 0$ such that $(n-1) \geq c_1(n-1)^3$.
   Step: We will show the proof is correct for n:
   $$n = n-1+1 \geq_{(1)} c_1(n-1)^3+1 \geq c_1 \cdot (n-1)^3 \geq_{(2)} c_1 \cdot (n-n/2)^3 = c_1(n/2)^3 = \frac{c_1}{8} \cdot n^3,$$
   where (1) is due to the induction assumption, and (2) is since for any $n \geq 2:\ n-1 \geq n-n/2$. Hence, taking $c = c_1/8$ we have that $n \geq c \cdot n^3$.

3. **Claim:** $n \in O(1)$.
   Proof: We will prove by induction on n that for every $n \geq 1$ it holds that $n \in O(1)$.
   Base: For $n = 1$ clearly $1 \in O(1)$.
   Assumption: assume the claim holds for $n-1$, meaning that $n-1 \in O(1)$.
   Step: We next show correctness for n:
   $$n = n-1+1 =_{(1)} O(1) + 1 = O(1) + O(1) = O(1),$$
   where (1) is due to the induction assumption.

1. **Claim:** $8n^2 \notin O(n^2)$.
   Proof: For every $n \geq 1$ it holds that $8n^2 > n^2$. Hence using $n_0 = 1, c = 1$ we conclude that $8n^2 \notin O(n^2)$.

The proof is incorrect

In order to prove the claim, we need to prove that there is no exist $n_0, c > 0$ s.t for any $n \geq n_0$ $8n^2 \leq c \cdot n^2$. And in this case they show only for $c = 1$.

2. **Claim:** $n \in \Omega(n^3)$.
   Proof: We will prove by induction on n, that there exists a constant $c > 0$ such that for every $n \geq 2$ it holds that $n \geq c \cdot n^3$. This is enough in order to show $n \in \Omega(n^3)$.
   Base: for $n = 2$, for $c = 1/4$ it holds that $2 = \frac{1}{4} \cdot 8 = c \cdot 2^3$.
   Assumption: Assume that the claim is correct for $n-1$, meaning there exists $c_1 > 0$ such that $(n-1) \geq c_1(n-1)^3$.
   Step: We will show the proof is correct for n:
   $$n = n-1+1 \geq_{(1)} c_1(n-1)^3+1 \geq c_1 \cdot (n-1)^3 \geq_{(2)} c_1 \cdot (n-n/2)^3 = c_1(n/2)^3 = \frac{c_1}{8} \cdot n^3,$$
   where (1) is due to the induction assumption, and (2) is since for any $n \geq 2:\ n-1 \geq n-n/2$. Hence, taking $c = c_1/8$ we have that $n \geq c \cdot n^3$.

The proof is incorrect

In order to prove that $n \in \Omega(n^3)$ we need to show that there exist $n_0, c > 0$ s.t for any $n > n_0$ $n \geq c \cdot n^3$. In this case they prove that for any n there exist c_1, which is may be different for each n.

3. **Claim:** $n \in O(1)$.
Proof: We will prove by induction on n that for every $n \geq 1$ it holds that $n \in O(1)$.
Base: For $n = 1$ clearly $1 \in O(1)$.
Assumption: assume the claim holds for $n-1$, meaning that $n-1 \in O(1)$.
Step: We next show correctness for n:

$$n = n - 1 + 1 =_{(1)} O(1) + 1 = O(1) + O(1) = O(1),$$

where (1) is due to the induction assumption.

The proof is incorrect

In order to prove that $n \in O(1)$ we need to show that there exist $n_0, c > 0$ s.t for any $n \geq n_0$ we get $n < c \cdot 1$. Since there will be always n s.t $n > c$. Therefore it does not meet the requirement.

Question C

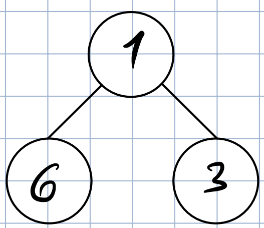
Mark True / False and justify your answer.

1. When implementing a min-heap using an array, the array representing the heap is sorted.
2. The smallest element in a max-heap with distinct elements must be a leaf.
3. In a min-heap with $n \geq 8$ elements, the right child of the root can be larger than $n/4$ of the elements in the heap.

1. When implementing a min-heap using an array, the array representing the heap is sorted.

False

Let $a = [1, 6, 3]$ be an array representing the following heap



We can see that the heap is min-heap but the array isn't sorted.

2. The smallest element in a max-heap with distinct elements must be a leaf.

True

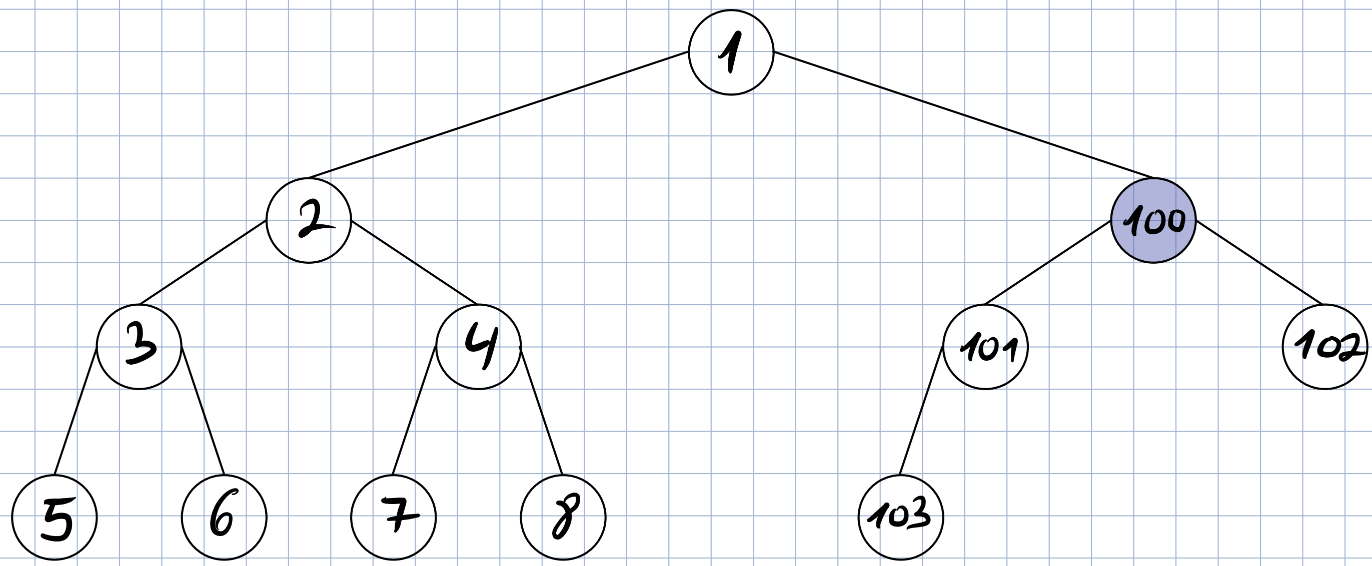
By definition, each element in max-heap is smaller than it children. A leaf is an element with no children.
Assume toward contradiction that the smallest element is not a leaf. Thus it has a child, by definition of max-heap the child is bigger than the element. Therefore it's not the smallest element. That is, the smallest element in max-heap is a leaf.

⑥

3. In a min-heap with $n \geq 8$ elements, the right child of the root can be larger than $n/4$ of the elements in the heap.

True

Example to min-heap that meet the condition:



We can see that its a valid min-heap with 12 elements.
The right child of the root is 100, which is bigger than 7 elements that are more than $\frac{1}{4}$ of the heap elements.

Question D

Prove or disprove:

1. Given that the best case complexity of the algorithm A is $O(f(n))$ and the worst case complexity of A is $\Omega(g(n))$. It follows that $f(n) \in \Omega(g(n))$.
2. Given that the worst case complexity of the algorithm A is $O(f(n))$ and the best case complexity of A is $\Omega(g(n))$. It follows that $f(n) \in \Omega(g(n))$.
3. Given that the average case complexity of the algorithm A is $\Theta(f(n))$ and the worst case complexity of A is $O(g(n))$. It follows that $f(n) \in O(g(n))$.

1. Given that the best case complexity of the algorithm A is $O(f(n))$ and the worst case complexity of A is $\Omega(g(n))$. It follows that $f(n) \in \Omega(g(n))$.

False

Let A_B be the best case and A_W be the worst case.
Let $f(n) = n$ and $g(n) = n^2$.
Assume $A_B \in O(f(n))$ and $A_W \in \Omega(g(n))$. Therefore:
For any $c_1, n > n_1$ it holds that $A_B \leq c_1 \cdot f(n) = c_1 \cdot n$
For any $c_2, n > n_2$ it holds that $A_W \geq c_2 \cdot g(n) = c_2 \cdot n^2$.
Assume toward contradiction that $f(n) \in \Omega(g(n))$. That is, there exist c, n_0 s.t for any $n > n_0$ $f(n) \geq c \cdot g(n)$, meaning $n > c \cdot n^2$. Since $c, n > 0$ we get that the inequality holds only for $n = c = 1$. That is n is bounded and the case

in the question does not hold.

2. Given that the worst case complexity of the algorithm A is $O(f(n))$ and the best case complexity of A is $\Omega(g(n))$. It follows that $f(n) \in \Omega(g(n))$.

True

Let A_B be the best case and A_W be the worst case.

Assume $A_W \in O(f(n))$ and $A_B \in \Omega(g(n))$, Therefore

For any $c_1, n > n_1$ it holds that $A_W \leq c_1 \cdot f(n)$.

For any $c_2, n > n_2$ it holds that $A_B \geq c_2 \cdot g(n)$.

Since $A_W \geq A_B$ we'll choose $n_0 = \max\{n_1, n_2\}$. Thus both of the condition of $O(f(n))$ and $\Omega(g(n))$ are met. Therefore, $c_1 \cdot f(n) \geq A_W \geq A_B \geq c_2 \cdot g(n)$. That is $c_1 \cdot f(n) \geq c_2 \cdot g(n)$. Meaning $f(n) \geq \frac{c_2}{c_1} \cdot g(n)$. That is for $c = \frac{c_2}{c_1}$ we get $f(n) \geq c \cdot g(n)$ for any $n \geq n_0$. Therefor $f(n) \in \Omega(g(n))$.

3. Given that the average case complexity of the algorithm A is $\Theta(f(n))$ and the worst case complexity of A is $O(g(n))$. It follows that $f(n) \in O(g(n))$.

True

Let A_V be the average case and A_W be the worst case.

Assume $A_W \in O(g(n))$ and $A_V \in \Theta(f(n))$, Therefore

For any $c_1, n > n_1$ it holds that $A_V \leq c_1 \cdot f(n)$

For any $c_2, n > n_2$ it holds that $A_V \geq c_2 \cdot f(n)$

For any $c_3, n > n_3$ it holds that $A_W \leq c_3 \cdot g(n)$

Since $A_W \geq A_V$ we'll choose $n_0 = \max\{n_1, n_3\}$. Thus both of the condition of $O(g(n))$ and $\Theta(f(n))$ are met. Therefore, $c_3 \cdot g(n) \geq A_W \geq A_V \geq c_1 \cdot f(n)$. That is $c_3 \cdot g(n) \geq c_1 \cdot f(n)$. Meaning $g(n) \geq \frac{c_1}{c_3} \cdot f(n)$. That is for $c = \frac{c_2}{c_1}$ we get $g(n) \geq c \cdot f(n)$ for any $n \geq n_0$. Therefor $f(n) \in O(g(n))$.

8

Question E

Define the following sequence

$$T(n) = \begin{cases} 3 & \text{if } n = 1, \\ T(n-1) + 3^n & \text{otherwise.} \end{cases}$$

Find a closed-form for $T(n)$ (no sigmas) and prove its correctness by induction.

$$T(n) = \frac{3 \cdot (3^n - 1)}{2}$$

We'll prove by induction that for every $n > 0$.

$$T(n) = \frac{3 \cdot (3^n - 1)}{2}.$$

Base case ($n=1$): $T(1) = \frac{3 \cdot 3^1 - 1}{2} = \frac{3 \cdot (3-1)}{2} = \frac{3 \cdot \cancel{2}}{\cancel{2}} = 3$. As in definition of T. Thus, the claim hold for $n=1$.

Induction hypothsis: Assume that for any $k > 1, k \in \mathbb{N}$, $n = k$. That is,

$$T(k) = \frac{3 \cdot (3^k - 1)}{2}.$$

Induction step: We'll prove the claim holds for $n = k+1$.

$T(k+1) = T(k) + 3^{k+1}$ By the induction assumption we get

$$T(k) + 3^{k+1} = \frac{3 \cdot (3^k - 1)}{2} + 3^{k+1} = \frac{3^{k+1} - 3}{2} + 3^{k+1} = \frac{3^{k+1} - 3}{2} + \frac{2 \cdot 3^{k+1}}{2} =$$

$$\frac{3 \cdot 3^{k+1} - 3}{2} = \frac{3 \cdot (3^{k+1} - 1)}{2}$$

Therefore the claim holds for $n = k+1$.

That is, by induction the claim holds for any $n > 0$.