**CONCORDIA UNIVERSITY**

**DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING**

**COMP 6651: Algorithm Design Techniques**

**Winter 2022**

**Quiz # 1**

### Question 1

What is the complexity for computing the following sum: $1^3 + 2^3 + \cdots + n^3$:

☐ $\Omega(n^2)$

☐ $\theta(n^4)$

☐ $\Omega(n^3)$

☐ $O(n^2)$

Solution

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

The first part of the identity is sometimes called Nicomache's theorem, see Figure 1 for an illustration of it.

Answers are therefore as follows:

☑ $\Omega(n^2)$

☑ $\theta(n^4)$

☑ $\Omega(n^3)$

☒ $O(n^2)$

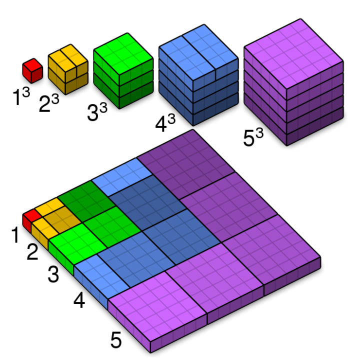


Figure 1: Visualisation graphique de l'égalité

**Question 2**

Assume that algorithm A and another algorithm B take $\log_2 n$ and $\sqrt{n}$ microseconds, respectively, to solve a problem. What is the largest size $n$ of a problem these algorithms can solve, respectively, in one second?

☐ $2^{10^6}$ and $10^6$

☐ $2^{10^6}$ and $10^{12}$

☐ $2^{10^6}$ and $6 \times 10^6$

☑ $2^{10^6}$ and $6 \times 10^{12}$

☐ $2^{10^3}$ and $3 \times 10^6$

☐ $2^{10^3}$ and $3 \times 10^3$

☐ $2^{10^3}$ and $6 \times 10^6$

☐ $2^{10^3}$ and $6 \times 10^3$

Solution

A microsecond is $10^{-6}$ seconds. Hence, a second $= 10^6$ microseconds.

One hour $= 60 \times 60 \times 10^6 = 3.6 \times 10^9$ microseconds.

One month (30 days) $= 2.592 \times 10^{12}$ microseconds.

One century $= 3.1104 \times 10^{15}$ microseconds.

$f(n) = \log n$ in this case, the largest value $n$ such that $\log_2 n \leq 10^6$.

We rewrite as, $2^{\log n} \leq 2^{10^6}$, thus $\log_2(n)$ and $\sqrt{n}$, in one second

$2^{10^6}$ and $10^{12}$ the value will be $10^6$ and $10^6$

**Question 3**

Consider the following algorithm

Integer $j, n$
While $j \leq n$
    $\leftarrow j \times j$

Assume initial value of $j$ such that $j \geq 2$ The number of comparisons made in the execution of the loop for any $> 0$ is

- ☐ $\lfloor \log_2 n \rfloor \times \log n$
- ☐ $n$
- ☐ $\lfloor \log_2 n \rfloor$
- ☑ $\lfloor \log_2 n \rfloor + 1$

Solution

Assume $n$ is a power of 2

For every iteration of while loop, velue of $j$ becomes half, hence after $\log n$ comparisons, it will become 1.

There is the need of a last comparison to terminate the whole loop

Hence total number of comparisons $= \log n + 1$

If number is not the power of 2, then we have to take the floor of $\log n$

Hence $\lfloor \log_2 n \rfloor + 1$ is the correct answer

**Question 4**

What is the solution to the recurrence relation $T(n) = 9T(n/3) + O(n)$ ?

☐ $n^2$

☐ $n^{\log_2 9}$

☑ $O(n^2)$

☑ $O(n^{\log_2 9})$

Solution

$n^2 = n^{\log_2 9}$ !!!