Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Algorithms and Complexity
Assignment 0

Name: Evangelos Argyropoulos AM: 1115202200010

Exercise 1

A

To order the functions I am using the complexity classes of $log_2(a_i)$, so only in the case where I end up with the same complexity for some $log_2(a_i)$, I need compare the corresponding a_i to order them.

1.

$$\log(a_1) = \log\left(\sqrt{n!}\right) = \frac{1}{2}\log(n!) = \Theta(n\log n)$$

because:

$$\left.\begin{array}{l}
\bullet\ n! = 1 \cdot 2 \cdot \ldots \cdot n \leq n \cdot n \cdot \ldots \cdot n = n^n \Rightarrow \log(n!) \leq \log(n^n) = n\log(n) \\
\Rightarrow \log(a_1) = O(n\log(n)) \\
\bullet\ n! = 1 \cdot 2 \cdot \ldots \cdot (n-1) \cdot n \geq \frac{n}{2} \cdot \left(\frac{n}{2}+1\right) \cdot \ldots \cdot n \geq \frac{n}{2} \cdot \frac{n}{2} \cdot \ldots \cdot \frac{n}{2} = \left(\frac{n}{2}\right)^{\frac{n}{2}} \\
\Rightarrow \log(n!) \geq \log\left(\left(\frac{n}{2}\right)^{\frac{n}{2}}\right) = \frac{n}{2}\log\left(\frac{n}{2}\right) = \frac{n}{2}\log n - \frac{n}{2} \\
\Rightarrow \log(a_1) = \Omega(n\log n)
\end{array}\right\} \Rightarrow \log(a_1) = \Theta(n\log n)$$

2.

$$a_2 = \binom{n}{6} = \frac{n!}{(n-6)!6!} = \frac{n(n-1)(n-2)(n-3)(n-4)(n-5)}{6!}$$

$$log(a_2) = log\left(\frac{n!}{(n-6)!6!}\right) = log\left(\frac{n(n-1)(n-2)(n-3)(n-4)(n-5)}{6!}\right)$$

$$\approx \log\left(n^6\right) = 6\log n = \Theta(\log n)$$

3.

$$log(a_3) = log(log^{10}(n)) = log((logn)^{10}) = log(10 \cdot logn) = log(10) + log(logn) = \Theta(log(logn))$$

4.

$$log(a_4) = log(n \cdot 2^n) = log(n) + log(2^n) = logn + n = \Theta(n)$$

5.

$$log(a_5) = log(log(5 \cdot n!)) = log(log5 + log(n!)) \overset{\text{ignoring log5 because log(n!) is increasing more}}{\approx} log(log(n!))$$

Using the basic arrangement:
$c < \log^* n < \log(\log(n)) < \log(n) < n^{0.5} < n < n\log(n) < n^p < a^n < n! < n^n, c \in \mathbb{R}, p \geq 2, a > 1$
there is:
$2^n < n! < n^n \overset{\log n \nearrow}{\Rightarrow} \log 2^n < \log n! < \log n^n \Rightarrow n < \log n! < n \log n \overset{\log n \nearrow}{\Rightarrow} \log n < \log(\log n!) < \log(n \log n) \Rightarrow \underline{\Theta(\log n) = \log n < \log(\log n!) < \log n + \log(\log n) = \Theta(\log n) \Rightarrow}$
$\underline{\Rightarrow \log(\log n!) = \Theta(\log n)}$

So, current order is: $a_3 < a_2, a_5 < a_4 < a_1$. Now we have to define the order between a_2 and a_5.

Comparing a_2 and a_5:

$$a_2 = \binom{n}{6} = \frac{n!}{(n-6)!6!} = \frac{n(n-1)(n-2)(n-3)(n-4)(n-5)}{6!} = \Theta(n^6)$$
$$a_5 = log(5 \cdot n!) = log(5) + log(n!) = \Theta(n \log n)$$

So, $a_5 < a_2$ (because $n^5 > \log n \Rightarrow n^6 > n \log n$).

Final order is $\underline{a_3 < a_5 < a_2 < a_4 < a_1}$

B

To order the functions I am using the basic-known order $c < \log^* n < \log(\log(n)) < \log(n) < n^{0.5} < n < n\log(n) < n^p < a^n < n! < n^n, c \in \mathbb{R}, p \geq 2, a > 1$
So, there is:
$b_3 = e^n = \Theta(e^n)$
$b_2 = 5n^7 = \Theta(n^7)$
$b_4 = log(n^5) = 5log(n) = \Theta(logn)$
So, $b_4 < b_2 < b_3$

Now I have to order $b_1 = \frac{n}{logn} = \Theta(\frac{n}{logn})$ and $b_5 = \frac{logn}{3n} = \Theta(\frac{logn}{n})$
Firstly, I will order b_1:

$$\lim_{n\to+\infty} \frac{b_2}{b_1} = \lim_{n\to+\infty} \frac{5n^7}{\frac{n}{\log n}} = \lim_{n\to+\infty} 5n^6 logn = +\infty \Rightarrow b1 = O(b_2)$$

$$\lim_{n\to+\infty} \frac{b_4}{b_1} = \lim_{n\to+\infty} \frac{\log(n^5)}{\frac{n}{\log n}} = \lim_{n\to+\infty} \frac{5\log^2 n}{n} \overset{\infty/\infty}{=} \lim_{n\to+\infty} \frac{10\log n}{n} \overset{\infty/\infty}{=} \lim_{n\to+\infty} \frac{10}{n} = 0 \Rightarrow b1 = \Omega(b_4)$$

So, $b_4 < b_1 < b_2 < b_3$. Now, I will order b_5.

$$\lim_{n\to+\infty} \frac{b_4}{b_5} = \lim_{n\to+\infty} \frac{log(n^5)}{\frac{logn}{3n}} = \lim_{n\to+\infty} \frac{15n\cancel{logn}}{\cancel{logn}} = +\infty \Rightarrow b_5 = O(b_4)$$

<u>**Final order is:** $b_5 < b_4 < b_1 < b_2 < b_3$</u>

C

To order the functions I am using the complexity classes of $log_2(c_i)$, so only in the case where I end up with the same complexity for some $log_2(c_i)$, I need compare the corresponding c_i to order them.

1. $$\log(c_1) = \log(n!) = \Theta(n \log n)$$

2. $$\log(c_2) = \log\left(2^{2^{2^n}}\right) = 2^{2^n} \log(2) = 2^{2^n} = \Theta(2^{2^n})$$

3. $$\log(c_3) = \log(8n) = \log\left(2^3\right) + \log n = 3 + \log n = \Theta(\log n)$$

4. $$c_4 = n^{\log(\log n)} = (\log n)^{\log n} \Rightarrow \log(c_4) = \log\left((\log n)^{\log n}\right) = \log n \cdot \log(\log n) = \Theta(\log n \cdot \log(\log n))$$

5. $$c_5 = 2n^8 + n^{\log n} \overset{(n^{\log n} \textit{ is increasing more})}{\Rightarrow} \log(c_5) \approx \log\left(n^{\log n}\right) = \log^2 n = \Theta(\log^2 n)$$

<u>Comparing $\log n, \log(\log n)$</u>

For significant large n $(n > 2^2)$

For $n > 2^2 \Rightarrow \log n > \log 2^2$

$\Rightarrow \log(\log n) > \log(2 \log 2)$

$\Rightarrow \log(\log n) > 1$

$\Rightarrow \log n \cdot \log(\log n) > \log n$ *[For $n > 2^2 \Rightarrow \log n > 2 > 0$]*

$\Rightarrow \log(c_4) > \log(c_3)$

<u>Comparing $\log n \cdot \log(\log n), \log^2 n$</u>

$n > \log n \Rightarrow \log n > \log(\log n)$

$\overset{(\text{for } n>2 \Rightarrow \log n > 1)}{\Rightarrow} \log n \cdot \log n > \log n \cdot \log(\log n) \Rightarrow \log^2 n > \log n \cdot \log(\log n)$

$\Rightarrow \log(c_5) > \log(c_4)$

So, we have the following arrangement:
$\log n < \log n \cdot \log(\log n) < \log^2 n < n \log n < 2^{2^n}$
And so,
<u>**Final order is:** $c_3 < c_4 < c_5 < c_1 < c_2$</u>

Exercise 2

(a.) **True**

$6n^2 + 3n - 6 = \Omega(7n^2) \iff \exists n_0 \in \mathbb{N}, \exists c \in \mathbb{R}^+ : (\forall n)\ 6n^2 + 3n - 6 \geq c7n^2$

So, we will show that $\lim\limits_{n\to\infty} \dfrac{6n^2+3n-6}{7n^2} \geq c$ for some constant $c > 0$

$\lim\limits_{n\to\infty} \dfrac{6n^2+3n-6}{7n^2} = \lim\limits_{n\to\infty} \dfrac{6+\frac{3}{n}-\frac{6}{n^2}}{7} = \dfrac{6}{7}$

Since $\dfrac{6}{7}$ is a positive constant, we can choose $c = \dfrac{6}{7}$

Therefore, we have shown that $6n^2 + 3n - 6 = \Omega(7n^2)$.

(b.) **False**

For instance, if f(n) = n and g(n) = n^2

Then, $(f+g)(n) = n + n^2$

but $f(n) = \Theta(n), g(n) = \Theta(n^2)$ and $(f+g)(n) = \Theta(n^2) \neq \Theta(n) = \Theta(min(f(n), g(n)))$

(c.) **True**

$\omega(f(n)) \cap o(f(n)) =$

$= \{g(n) \,|\, \forall c > 0\, \exists n_0 \in \mathbb{N}\, \forall n \geq n_0 : g(n) < cf(n)\} \cap \{g(n) \,|\, \forall c > 0\, \exists m_0 \in \mathbb{N}\, \forall m \geq m_0 : g(n) > cf(n)\}$

$= \{g(n) \,|\, \forall c > 0\, \exists k_0 \in \mathbb{N}\, \forall k \geq k_0 : g(n) < cf(n) \wedge g(n) > cf(n)\} = \emptyset$

Otherwise, I can prove it using the definitions taught in the course:
$f(n) = \omega(g(n)) \Rightarrow \nexists(C,k) : f(n) = O(g(n)) \Rightarrow \nexists(C,k) : f(n) = o(g(n))$. So, $\omega(f(n)) \cap o(f(n)) = \emptyset$

(d). **False**
For instance, if $g(n) = \left(\frac{1}{2}\right)^n$, then:

$$2024 \cdot g(n) + 2^{2024} = 2024 \cdot \left(\frac{1}{2}\right)^n + 2^{2024} = 2024 \cdot 2^{-n} + 2^{2024} = O(1) \neq O\left(\left(\frac{1}{2}\right)^n\right)$$

$$\mathcal{L} = \lim_{n\to\infty} \frac{2024g(n) + 2^{2024}}{g(n)} = \lim_{n\to\infty} \left(2024 + \frac{2^{2024}}{\left(\frac{1}{2}\right)^n}\right) = \lim_{n\to\infty} \left(2024 + 2^{2024-(-n)}\right) =$$
$$= \lim_{n\to\infty} \left(2024 + 2^{2024+n}\right) = +\infty$$

So, I can infer that:
$\mathbb{L} = \infty \Rightarrow 2024g(n) + 2^{2024} = \omega(g(n)) \Rightarrow \nexists(C,k) : 2024g(n) + 2^{2024} = O(g(n))$

(e.) **False**
[if I get $\mathcal{L} = \lim_{n\to\infty} \frac{n+2024\sqrt{n}}{n\sqrt{n}}$*, then*
$n + 2024\sqrt{n} = \Omega(n\sqrt{n}) \iff 0 < \mathcal{L} \leq \infty$*]*

$$\mathcal{L} = \lim_{n\to\infty} \frac{n + 2024\sqrt{n}}{n\sqrt{n}} = \lim_{n\to\infty} \left(\frac{1}{\sqrt{n}} + \frac{2024}{n}\right) = 0 \Rightarrow n + 2024\sqrt{n} \notin \Omega(n\sqrt{n})$$

Exercise 3

A	B	O	o	Ω	ω	Θ
$\left(\frac{4}{5}\right)^n$	$\log n$	**Yes**	**Yes**	No	No	No
$n^4 - 3n^3$	$16^{\log n}$	**Yes**	No	**Yes**	No	**Yes**
$\sum_{k=0}^{n} \binom{n}{k}$	$2^{\frac{n}{3}}$	No	No	**Yes**	**Yes**	No
$\sum_{k=1}^{n} (k + \log k)$	$n^2 \log n$	**Yes**	**Yes**	No	No	No
$\binom{n}{n-3}$	$2024n^3$	**Yes**	No	**Yes**	No	**Yes**

<u>Proof:</u>

- $\left(\frac{4}{5}\right)^n$ —— $\log n$

$$0 < \frac{4}{5} < 1 \Rightarrow \left(\frac{4}{5}\right)^n \searrow \mathbb{R} \text{ while } \log n \nearrow (0, +\infty)$$

$\{\log n\}$ *is an increasing function in contrast with* $\{\left(\frac{4}{5}\right)^n\}$ *which is decreasing,*

so it is obvious that $\left(\frac{4}{5}\right)^n = o(\log n)$ *(and of course* $\left(\frac{4}{5}\right)^n = O(\log n)$*)*

because $\lim_{n\to\infty} \left(\frac{4}{5}\right)^n = 0$ *and* $\lim_{n\to\infty} (\log n) = +\infty \Rightarrow \{\exists n_0 : \forall n \geq n_0 \ \log n > \left(\frac{4}{5}\right)^n\}$

I will also prove it with the mathematical definitions using the corresponding limit:

$$\lim_{n\to\infty} \frac{\left(\frac{4}{5}\right)^n}{\log n} = \lim_{n\to\infty} \left(\frac{4}{5}\right)^n \cdot \frac{1}{\log n} = 0 \cdot 0 = 0 \Rightarrow \left(\frac{4}{5}\right)^n = o(\log n) \wedge \left(\frac{4}{5}\right)^n = O(\log n).$$

- $n^4 - 3n^3$ —— $16^{\log n}$

  There is:
  $n^4 - 3n^3 = \Theta(n^4)$
  $16^{\log n} = n^{\log 16} = n^{\log 2^4} = n^{4 \log 2} = n^4 = \Theta(n^4)$
  Obviously $n^4 - 3n^3 = \Theta(16^{\log n}(= n^4))$
  $$\lim_{n\to\infty} \frac{n^4 - 3n^3}{16^{\log n}} = \lim_{n\to\infty} \frac{n^4 - 3n^3}{n^{\log 2^4}} = \lim_{n\to\infty} \frac{n^4 - 3n^3}{n^4} = 1$$
  From definitions also is implied that $n^4 - 3n^3 = \Theta(16^{\log n}) = O(16^{\log n}) \cap \Omega(16^{\log n})$

- $\sum_{k=0}^{n} \binom{n}{k}$ —— $2^{\frac{n}{3}}$

  In accordance with the binomial theorem, there is: $(a+b)^n = \displaystyle\sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k$

  So, for $a = b = 1$, there is $\displaystyle\sum_{k=0}^{n} \binom{n}{k} 1^{n-k} 1^k = (1+1)^n \Rightarrow \sum_{k=0}^{n} \binom{n}{k} = 2^n$.

  $$\lim_{n\to\infty} \frac{\sum_{k=0}^{\infty} \binom{n}{k}}{2^{\frac{n}{3}}} = \lim_{n\to\infty} \frac{2^n}{2^{\frac{n}{3}}} = \lim_{n\to\infty} 2^{n-\frac{n}{3}} = \lim_{n\to\infty} 2^{\frac{2n}{3}} = +\infty$$

  So, $\sum_{k=0}^{n} \binom{n}{k} = \Omega(2^{\frac{n}{3}})$ and $\sum_{k=0}^{n} \binom{n}{k} = \omega(2^{\frac{n}{3}})$

- $\sum\limits_{k=1}^{n}(k+\log k)$ —— $n^2\log n$

$$\sum_{k=1}^{n}(k+\log k)=\sum_{k=1}^{n}(k)+\sum_{k=1}^{n}(\log k)=\frac{n(n+1)}{2}+\log 1+\log 2+...+\log(n-1)+\log n$$
$$=\frac{n(n+1)}{2}+\log(1\cdot 2\cdot ..\cdot(n-1)\cdot n)=\frac{n(n+1)}{2}+\log(n!)$$

Also, I can infer the following two:

$$\to \frac{n(n+1)}{2}=\frac{1}{2}n^2+\frac{1}{2}n=\Theta(n^2)$$
$$\to \log n! = \Theta(n\log n)$$

(the first $\left(\frac{n(n+1)}{2}\right)$ is obvious and the second ($\log n!$) is proven in previous exercises.)

Also, $n > \log n \Rightarrow n^2 > n\log n$. So, $\sum\limits_{k=1}^{n}(k+\log k)=\Theta(n^2)$.

Now we have that, for $n > 2 \Rightarrow \log n > 1 \Rightarrow n^2\log n > n^2 \Rightarrow$

$$\Rightarrow \sum_{k=1}^{n}(k+\log k)=o(n^2\log n)\wedge\sum_{k=1}^{n}(k+\log k)=O(n^2\log n)$$

Also, because $\sum\limits_{k=1}^{n}(k+\log k)=o(n^2\log n)$*, it cannot be neither* $\Omega(n^2\log n)$*,*
nor $\omega(n^2\log n)$*, nor* $\Theta(n^2\log n)$

- $\binom{n}{n-3}$ —— $2024n^3$

$$\binom{n}{n-3}=\frac{n!}{(n-3)!3!}=\frac{n(n-1)(n-2)}{3!}=\Theta(n^3)$$
$$2024n^3=\Theta(n^3)$$

So, $\binom{n}{n-3}=\Theta(2024n^3)\Rightarrow\binom{n}{n-3}=O(2024n^3)\wedge\binom{n}{n-3}=\Omega(2024n^3)$

I will also prove it using a limit:

$$\lim_{n\to\infty}\frac{\binom{n}{n-3}}{2024n^3}=\lim_{n\to\infty}\frac{\frac{n!}{(n-3)!3!}}{2024n^3}=\lim_{n\to\infty}\frac{\frac{n(n-1)(n-2)}{3!}}{2024n^3}=\lim_{n\to\infty}\frac{n^3-3n^2+2n}{2024\cdot 3!\cdot n^3}=\lim_{n\to\infty}\frac{n^3(1-\frac{3}{n}+\frac{2}{n^2})}{2024\cdot 3!\cdot n^3}=$$
$$=\lim_{n\to\infty}\frac{(1-\frac{3}{n}+\frac{2}{n^2})}{2024\cdot 3!}=\frac{1}{2024\cdot 3!}\in\mathbb{R}^*\Rightarrow$$
$$\Rightarrow\binom{n}{n-3}=\Theta(2024n^3)\wedge\binom{n}{n-3}=O(2024n^3)\wedge\binom{n}{n-3}=\Omega(2024n^3)$$

Exercise 4

Algorithm 1

1st for: n steps $\Rightarrow O(n)$
2nd for: i steps with $max_i = n \Rightarrow O(n)$
Calculation of arg-CALC function:
I observe that s is increasing with a rate 1 + 2 + 3 + ... + i until $s > m$
$\Rightarrow \sum_{i=1}^{s_i}(i) > m \Rightarrow \frac{s_i(s_i+1)}{2} > m \Rightarrow s_i^2 + s_i > 2m \Rightarrow s_i^2 + s_i + \frac{1}{4} > 2m + \frac{1}{4} \Rightarrow$
$(s_i + \frac{1}{2})^2 > 2m + \frac{1}{4}\ (> 0) \Rightarrow s_i + \frac{1}{2} > \sqrt{2m + \frac{1}{4}} \Rightarrow s_i > \sqrt{2m + \frac{1}{4}} - \frac{1}{2} \Rightarrow$
The arg-CALC function does $O(\sqrt{n})$ steps, because
there is: $s_i > \sqrt{2m + \frac{1}{4}} - \frac{1}{2} \wedge max_m = max_j = i \wedge max_i = n \Rightarrow O(\sqrt{n})$

So, overall complexity is $O(n) \cdot O(n) \cdot O(\sqrt{n}) = \mathbf{O(n^2\sqrt{n})}$

Algorithm 2

1st for: n steps $\Rightarrow O(n)$
2nd for: In this loop, j is increasing with a rate $\underline{j_k \leftarrow j_{k-1} + 2j_{k-1} = 3j_{k-1},\ (1)}$ and so we can easily infer that the step of the loop increases at a geometric rate 3^j (It is easy also to observe that j gets the values $1(=3^0), 3(=3^1), 9(=3^2), 27(=3^3), 81(=3^4), ..., 3^k$).
So, the loop ends when:
$3^j > n \Rightarrow j > \log_3 n = \frac{1}{\log_2 3} \cdot \log_2 n \Rightarrow$ 2nd loop complexity is: $O(\log n)$

Calculating complexity of arg-CALC function:
It is easy to observe that the loop inside the function is executed $2m - m + 1 = m + 1$ times. This is because it starts from $s = m$ and ends at $s = 2m$ (including the iteration where s = 2m) with step = 1. Also, $max_m = max_j = n \Rightarrow$ max steps = n + 1, so the complexity of the procedure is: $O(n)$

Finally, the overall complexity of the algorithm is: $\mathbf{O(n) \cdot O(logn) \cdot O(n) = O(n^2 logn)}$

Exercise 5

The probability (p_{n+1}) to not find the element 'x' we are looking for is complementary to the sum of the probabilities of finding it in any of the n positions of the array.

$$p_{n+1} = 1 - \sum_{i=1}^{n} p_i = 1 - \sum_{i=1}^{\frac{n}{2}} \left(\frac{2}{3n}\right) - \sum_{i=\frac{n}{2}+1}^{n-2} \left(\frac{1}{3(n-4)}\right) - \sum_{i=n-1}^{n} \left(\frac{1}{4}\right) =$$

$$= 1 - \left(\frac{n}{2}\right)\left(\frac{2}{3n}\right) - \left(n - 2 - \frac{n}{2} - 1 + 1\right)\left(\frac{1}{3(n-4)}\right) - 2 \cdot \frac{1}{4} =$$

$$= 1 - \frac{1}{3} - \left(\frac{n}{2} - 2\right)\left(\frac{1}{3(n-4)}\right) - \frac{1}{2} = \frac{1}{2} - \frac{1}{3} - \frac{\cancel{n-4}}{6\cancel{(n-4)}} = 0$$

So, the expected value of the linear search in the array is:

$$\sum_{i=1}^{n}(i \cdot p_i) = \sum_{i=1}^{\frac{n}{2}} i \cdot \left(\frac{2}{3n}\right) + \sum_{i=\frac{n}{2}+1}^{n-2} i \cdot \left(\frac{1}{3(n-4)}\right) + \sum_{i=n-1}^{n} i \cdot \left(\frac{1}{4}\right) =$$

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

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

$$= \frac{n+2}{12} + \left(\frac{1}{3(n-4)}\right) \left(\frac{(n-2)(n-1)}{2} - \frac{\frac{n}{2}(\frac{n}{2}+1)}{2}\right) + \frac{2n-1}{4} =$$

$$= \frac{n+2}{12} + \left(\frac{1}{3(n-4)}\right) \left(\frac{n^2-3n+2}{2} - \frac{n^2+2n}{8}\right) + \frac{2n-1}{4} =$$

$$= \frac{n+2}{12} + \left(\frac{1}{3(n-4)}\right) \left(\frac{3n^2-14n+8}{8}\right) + \frac{2n-1}{4} =$$

$$= \frac{n+2}{12} + \frac{(3n-2)\cancel{(n-4)}}{24\cancel{(n-4)}} + \frac{2n-1}{4} =$$

$$= \frac{17n-4}{24} = O(n)$$