Homework 1 for Design and Analysis of Algorithms (Spring 2021)

Due: March 12th, 2021

Problem 1. (Exercise 0.1 in [DPV08]) In each of the following situations, indicate whether $f = O(g)$, or $f = \Omega(g)$, or both (in which case $f = \Theta(g)$).

	$f(n)$	$g(n)$
(a)	$n - 100$	$n - 200$
(b)	$n^{1/2}$	$n^{2/3}$
(c)	$100n + \log n$	$n + (\log n)^2$
(d)	$n \log n$	$10n \log 10n$
(e)	$\log 2n$	$\log 3n$
(f)	$10 \log n$	$\log(n^2)$
(g)	$n^{1.01}$	$n \log^2 n$
(h)	$n^2/\log n$	$n(\log n)^2$
(i)	$n^{0.1}$	$(\log n)^{10}$
(j)	$(\log n)^{\log n}$	$n/\log n$
(k)	$\sqrt{n}$	$(\log n)^3$
(l)	$n^{1/2}$	$5^{\log_2 n}$
(m)	$n2^n$	3^n
(n)	2^n	2^{n+1}
(o)	$n!$	2^n
(p)	$(\log n)^{\log n}$	$2^{(\log_2 n)^2}$
(q)	$\sum_{i=1}^{n} i^k$	n^{k+1}.

Solution.

(a) $f = \Theta(g)$

(b) $f = O(g)$

(c) $f = \Theta(g)$

(d) $f = \Theta(g)$

(e) $f = \Theta(g)$

(f) $f = \Theta(g)$

(g) $f = \Omega(g)$

(h) $f = \Omega(g)$

(i) $f = \Omega(g)$

(j) $f = \Omega(g)$

$$\begin{aligned} \log n &= \log_{\log n} n * \log\log n \\ f(n) &= (\log n)^{\log_{\log n} n * \log\log n} \\ &= n^{\log\log n} \\ &= \Omega(g(n)) \end{aligned}$$

(k) $f = \Omega(g)$

(l) $f = O(g)$

$$\begin{aligned} g(n) &= 5^{\log_2 n} = 2^{\log_2 5 \log_2 n} \\ &= n^{\log_2 5} = \Omega(f(n)) \\ f &= O(g) \end{aligned} \tag{1}$$

(m) $f = O(g)$

(n) $f = \Theta(g)$

(o) $f = \Omega(g)$

(p) $f = O(g)$

(q) k>-1 $f = \Theta(g)$, k=-1 $f = \Omega(g)$.

1 k>0

$$\begin{aligned} f(n) &< \int_1^{n+1} x^k dx = \Theta(n^{k+1}) \\ f(n) &> \int_0^n x^k dx = \Theta(n^{k+1}) \\ f(n) &= \Theta(g(n)) \end{aligned}$$

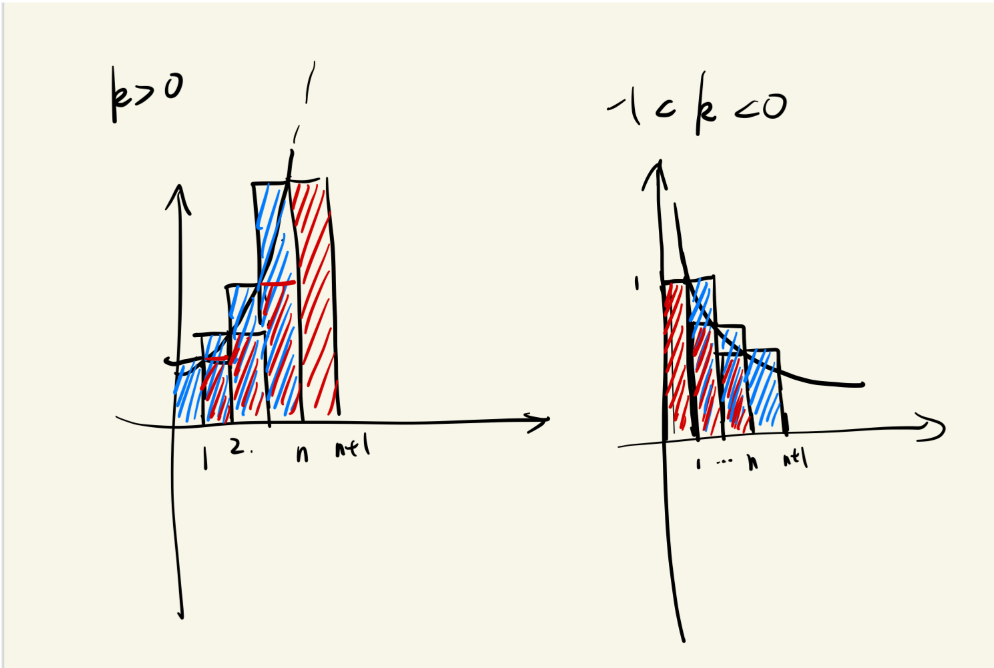
2 k=0 Obviously, $f(n) = \Theta(g(n))$

3 -1<k<0

$$\begin{aligned} f(n) &< 1 + \int_1^n x^k dx = \Theta(n^{k+1}) \\ f(n) &> \int_1^{n+1} x^k dx = \Theta(n^{k+1}) \\ f(n) &= \Theta(g(n)) \end{aligned}$$

4 k=-1 $f(n) = \Theta(\log n) = \Omega(g(n))$

5 k<-1 Both $f(n)$ and $g(n)$ are decreasing.



□

Problem 2. (Exercise 0.2 in [DPV08]) Show that, if c is a positive real number, then $g(n) = 1+c+c^2+\cdots+c^n$ is:

(a) $\Theta(1)$ if $c < 1$.

(b) $\Theta(n)$ if $c = 1$.

(c) $\Theta(c^n)$ if $c > 1$.

Problem 3. Show that there exists a C++ program P who can generate all pairs of natural numbers (x, y) such that $P_x(y)$ terminates [1]. That is, the program P can keep printing pairs of numbers (x, y) such that $P_x(y)$ terminates and for every (x', y') such that $P_{x'}(y')$ terminates, the program P can print it at some time.

Solution. □

[1] Since the number of such pairs are infinite, the program P must run forever.

Algorithm 1: Solution 3

```
F ← ∅ // pairs have been output
for S = 1 : +∞ do
    for T = 1 : S do
        for x = 0 : S − T do
            y = S − T − x
            Simulate P_x(y) for T steps // (or seconds, hours, ...)
            if P_x(y) terminates in T steps then
                if (x, y) ∉ F then
                    Output (x, y)
                    F ← F ∪ {(x, y)}
                end
            end
        end
    end
end
```

References

[DPV08] Sanjoy Dasgupta, Christos H Papadimitriou, and Umesh Virkumar Vazirani. *Algorithms*. McGraw-Hill Higher Education New York, 2008. 1, 3