Homework Assignment 1

CSE33101 Intro to Algorithms (Spring 2022)

Due: 2022-04-05 11:59 pm

Handwrite your answer to the following questions in English, scan it, and submit it to BlackBoard. **Illegible answers will not be graded** (zero points).

Total 10 points

1. Show that if $2^{n+1} = \Theta(2^n)$ or not by using the definition of O and Ω notation (2 points).

2. Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, \ldots, g_{26}$ of the functions satisfying $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \ldots, g_{25} = \Omega(g_{26})$. Partition your list into equivalence classes such that functions $f(n)$ and $g(n)$ are in the same class if and only if (iff) $f(n) = \Theta(g(n))$ (2 points).

$n \lg n$	$2^{2^{n+1}}$	$(\sqrt{2})^{\lg n}$	n^2	$n!$	$(\lg n)!$	$(\lg n)^{\lg n}$	e^n	$4^{\lg n}$
$(\frac{3}{2})^n$	n^3	$\lg^2 n$	$\lg(n!)$	2^{2^n}	$n^{1/\lg n}$	$(n+1)!$	$\sqrt{\lg n}$	$2^{\sqrt{2 \lg n}}$
$\ln \ln n$	2^n	$n \cdot 2^n$	$n^{\lg \lg n}$	$\ln n$	1	$2^{\lg n}$	n	

3. Let's define a sequence S_1, S_2, S_3, … by the rule that $S_1 = 1$, $S_2 = 1$, $S_3 = 2$ and every further term is the sum of the proceeding two. Thus, the sequence begins 1, 1, 2, 3, 5, 8, 13, … . If $k = (1+\sqrt{5})/2$, prove if the following is true or not for all positive integers n by using mathematical induction (2 points).

$$S_n \leq k^{n-1}$$

4. Prove that the number of different triples that can be chosen from n items is precisely $n(n-1)(n-2)/6$ by using mathematical induction (2 points).

5. Think of a process moving from the integer x to y via multiple steps based on the following rules.

   A. The length of each step is nonnegative, and the length of the first and last step is one.

   B. The length of the next step is either one less than, equal to, or one greater than the length of the previous step.

   For example, moving from the integer 10 to 15 takes four steps (i.e., 1+2+1+1). Write a pseudocode algorithm that finds the smallest number of steps when moving from the integer x to y (2 points).

Algorithm Assignment 1

20161190
이원재

1. Let's first use O definition

$O(g(n)) = \{ f(n) :$ there exist positive constants c and n_0 such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0 \}$

So when we assume $n_0 = 0$ $\quad 0 \le 2^{n+1} = 2 \times 2^n \le c \times 2^n$

We can find many constant c (example $c = 5$)

So, we can express $2^{n+1} = O(2^n)$

And then Let's use Ω definition

$\Omega(g(n)) = \{ f(n) :$ there exist positive constants c and n_0 such that $0 \le cg(n) \le f(n)$ for all $n \ge n_0 \}$

So when we assume $n_0 = 0$ $\quad 0 \le c \times 2^n \le 2^{n+1}$

We can find many constant c (example $c = 1$)

So, we can express $2^{n+1} = \Omega(2^n)$

We also learned that Theorem 2.1 For any two functions $f(n)$ ans $g(n)$, We have $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$

So, we can say $2^{n+1} = \Theta(2^n)$.

+ addly, the definition of Θ,

We can find the value of c_1, c_2, n_0 that satisfy

$0 \le c_2 2^n \le 2^n \times 2 \le c_1 2^n \Rightarrow 0 \le c_2 \le 2 \le c_1$

For example $n_0 = 1$, $c_1 = 12$, $c_2 = 1$ $\quad \therefore$ We also can say that

$2^{n+1} = \Theta(2^n)$

2. Let's classify one by one first!

- $n\lg n = \Theta(n\lg n)$
- $n^2 = \Theta(n^2)$
- $(\lg n)^{\lg n} = \Theta(\lg n^{\lg n})$
- $(\frac{3}{2})^n = \Theta(1.5^n)$
- $\lg(n!) = \Theta(n\lg n)$
- $(n+1)! = \Theta((n+1)!)$
- $\ln\ln n = \Theta(\ln(\ln n))$
- $\ln n = \Theta(\ln n)$
- $n = \Theta(n)$
- $2^{2^{n+1}} = \Theta(2^{2^{n+1}})$
- $n! = \Theta(n!)$
- $e^n = \Theta(2.7^n)$
- $n^3 = \Theta(n^3)$
- $2^{2^n} = \Theta(2^{2^n})$
- $\sqrt{\lg n} = \Theta(\lg n^{\frac{1}{2}})$
- $2^n = \Theta(2^n)$
- $1 = \Theta(1)$
- $(\sqrt{2})^{\lg n} = 2^{\frac{1}{2}\lg n} = \Theta(n^{\frac{1}{2}})$
- $(\lg n)! = \Theta((\lg n)^{\lg n + 1/2} e^{-\lg n})$
- $4^{\lg n} = \Theta(n^2)$
- $\lg^2(n) = \Theta(\lg^2(n))$
- $n^{\frac{1}{\lg n}} = n^{\frac{\lg 2}{\lg n}} = 2 = \Theta(1)$
- $2^{\sqrt{2\lg n}} = \Theta(2^{\sqrt{2\lg n}})$
- $n\cdot 2^n = \Theta(n\cdot 2^n)$
- $2^{\lg n} = \Theta(n)$

Rank above function

$$2^{2^{n+1}} > 2^{2^n} > (n+1)! > n! > e^n > n\cdot 2^n > 2^n >$$

$$\left(\tfrac{3}{2}\right)^n > \begin{matrix} n^{\lg\lg n} \\ (\lg n)^{\lg n} \end{matrix} > (\lg n)! > n^3 > \begin{matrix} n^2 \\ 4^{\lg n} \end{matrix} > \begin{matrix} n\lg n \\ \lg(n!) \end{matrix} > \begin{matrix} n \\ 2^{\lg n} \end{matrix} >$$

$$(\sqrt{2})^{\lg n} > 2^{\sqrt{2\lg n}} > \lg^2 n > \ln n > \sqrt{\lg n} > \ln\ln n > \begin{matrix} n^{\frac{1}{\lg n}} \\ 1 \end{matrix}$$

$\cancel{n\lg n}$	$\cancel{2^{2^{n+1}}}$	$\cancel{(\sqrt{2})^{\lg n}}$	$\cancel{n^2}$	$\cancel{n!}$	$\cancel{(\lg n)!}$	$\cancel{(\lg n)^{\lg n}}$	$\cancel{e^n}$	$\cancel{4^{\lg n}}$
$\cancel{(\frac{3}{2})^n}$	$\cancel{n^3}$	$\cancel{\lg^2 n}$	$\cancel{\lg(n!)}$	$\cancel{2^{2^n}}$	$\cancel{n^{1/\lg n}}$	$\cancel{(n+1)!}$	$\sqrt{\lg n}$	$\cancel{2^{\sqrt{2\lg n}}}$
$\cancel{\ln\ln n}$	$\cancel{2^n}$	$\cancel{n 2^n}$	$\cancel{n^{\lg\lg n}}$	$\cancel{\ln n}$	$\cancel{1}$	$\cancel{2^{\lg n}}$	$\cancel{n}$	

3.

- base case) when $n=1$ $\quad S_1^{(1)} \le 1 \quad (1 \le 1) \Rightarrow$ true.

  I check also $n=2$ $\quad S_2^{(1)} \le k = \frac{1+\sqrt{5}}{2} \Rightarrow$ true $\left(\frac{1+1}{2} < \frac{1+\sqrt{5}}{2}\right)$

- Inductive step)

  We assume $S_n \le k^{n-1}$ is true. We can say all positive integer n satisfy $S_n \le k^{n-1}$, then $S_{n+1} \le k^n$ also true.

  So we need to show $S_{n+1} \le k^n$

$$S_{n+1} = S_n + S_{n-1} \le k^{n-1} + k^{n-2} = k^{n-2}(k+1)$$

$$= \left(\frac{1+\sqrt{5}}{2}\right)^{n-2}\left(\frac{3+\sqrt{5}}{2}\right)$$

$$= \left(\frac{1+\sqrt{5}}{2}\right)^{n-2}\left(\frac{1+\sqrt{5}}{2}\right)^2$$

$$= \left(\frac{1+\sqrt{5}}{2}\right)^n \Rightarrow \therefore S_{n+1} \le k^n$$

by basic case and inductive step,

We can say all positive integer n satisfy that formula

$$S_n \le k^{n-1} \quad \left(k = \frac{1+\sqrt{5}}{2}\right)$$

4.

- Basic case) when $n=3$,

$$\frac{3(3-1)(3-2)}{6} = 1$$

(The number of different triples that can be choosen from 3 is 1)

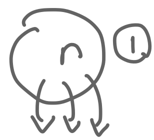
additional test when $n=4$

$$\frac{4(4-1)(4-2)}{6} = 4$$

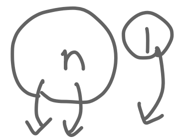
(" choosen from 4 is 4)

- Inductive step

we can think two case, when we choose 3 sample from $(n+1)$ sample.



<case 1>



<case 2>

Case 1) choose 3 things from n $\Rightarrow$ ${}_nC_3$

Case 2) choose 2 things from n and choose 1 thing from 1) $\Rightarrow$ ${}_nC_2$

So when we choose 3 things from $n+1$

$$= \frac{n(n-1)(n-2)}{6} + \frac{n(n-1)}{2} = \frac{n(n-1)(n-2)+3n(n-1)}{6} = \frac{(n-1)n(n+1)}{6}$$

So, if $n=k$ then $F(k) = \frac{k(k-1)(k-2)}{6}$, then

$$F(k+1) = \frac{(k-1)k(k+1)}{6}.$$

So number of different triples choosen from N item is

$$\frac{n(n-1)(n-2)}{6}$$

5.

First I get some answer by hand, and find this algorithm's rule.

2 ⇒ 11 ⇒ 2
3 ⇒ 11 1 ⇒ 3
4 ⇒ 11 2 ⇒ 3
5 ⇒ 11 21 ⇒ 4
6 ⇒ 11 22 ⇒ 4

7 ⇒ 11 22 1 ⇒ 5
8 ⇒ 11 22 2 ⇒ 5
9 ⇒ 11 22 3 ⇒ 5

10 ⇒ 11 22 31 ⇒ 6
11 ⇒ 11 22 32 ⇒ 6
12 ⇒ 11 22 33 ⇒ 6

13 ⇒ 11 22 33 1 ⇒ 7
14 ⇒ 11 22 33 2 ⇒ 7
15 ⇒ 11 22 33 3 ⇒ 7
16 ⇒ 11 22 33 4 ⇒ 7

17 ⇒ 11 22 33 41 ⇒ 8
18 ⇒ 11 22 33 42 ⇒ 8
19 ⇒ 11 22 33 43 ⇒ 8
20 ⇒ 11 22 33 44 ⇒ 8

21 ⇒ 11 22 33 44 1 ⇒ 9
22 ⇒ 11 22 33 44 2 ⇒ 9
23 ⇒ " 3 ⇒ 9
24 ⇒ " 4 ⇒ 9
25 ⇒ 11 22 33 44 5 ⇒ 9

26 ⇒ " 51 ⇒ 10
27 ⇒ 52 ⇒ 10
28 ⇒ 53 ⇒ 10
29 ⇒ 54 ⇒ 10
30 ⇒ 55 ⇒ 10

31 ~ 36 ⇒ 11
37 ~ 42 ⇒ 12
⋮

at the above study, I find rule that if distance increase one, and the result is same with (1, 2, 2, 3, 3, 4, 4, ...) and increase one.

$$2,\quad 2+2,\quad 2+2+2,\quad 2+2+2+3,\quad 2+2+2+3+3,\quad 2+2+3+3+4, \quad \dots$$
$$\Downarrow \quad\quad \Downarrow \quad\quad \Downarrow \quad\quad \Downarrow \quad\quad \Downarrow \quad\quad \Downarrow$$
$$2 \quad\quad 3 \quad\quad 4 \quad\quad 5 \quad\quad 6 \quad\quad 7$$

So I can express by function.

$$2 + 2(1+2+3+\dots+n-1) = b \text{ (distance)}$$

$$= 2 + 2\left(\frac{n(n+1)}{2} - 1\right) = n^2 + n = b = F(n)$$

distance ⇒ $F(1) = 1$, $F(2) = 6$, $F(3) = 12$, $F(4) = 20$

smallest number of step ⇒ $1 \times 2 = 2$, $2 \times 2 = 4$, $3 \times 2 = 6$, $4 \times 2 = 8$

Then, $n^2 + n + \frac{1}{4} = b + \frac{1}{4} \Rightarrow n = \sqrt{\frac{4b+1}{4}} - \frac{1}{2}$

So we can use distance (y) and can find smallest step (n) at the range of $k^2 \le 4y \le (k+1)^2$

So If we find the smallest value i that satisfy $4y \le i^2$ then $\sqrt{\frac{i^2}{4}} - \frac{1}{2} = \frac{i-1}{2} = \frac{n}{2}$

So $n = i-1$ (smallest step)

So, pseudocode is here

```
int d1, d2;    cin >> d1, d2;
int dist = abs(d1-d2);
int sol = 0;          // this is bound to range of input value
for (int i=1; i< 100000; i++) {
    if (i*i > 4*dist) {
        sol = i-1;
        break;
    }
}
cout << sol;
```

* I also attach real code at back page

Real code is also here.

test.cpp > main()

```cpp
#include <iostream>
#include <algorithm>
using namespace std;

int main(){
    cin.tie(0);
    cin.sync_with_stdio(0);

    int d1,d2;
    cin >> d1 >> d2;
    int dist = abs(d1-d2);
    cout << "dist is : " << dist << '\n';

    int flag = 0;
    for(int i=1; i< 1000000; i++){ // i< 범위는 dist 값에 따라 처리
        if(i*i >= 4 * dist){
            //cout << "right point is: " << i << '\n';
            flag = i-1; // consider sqrt(4y+1/4) - 1/2
            break;
        }
    }
    cout << "smallest number of step: " << flag;
}
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

```
dist is : 18
smallest number of step: 8%
(base) minjaelee@minjaeleeui-MacBookPro coding % ./test
3 10
dist is : 7
smallest number of step: 5%
(base) minjaelee@minjaeleeui-MacBookPro coding %
```