

Recitation 1 – Exercises from Chapter 1, 2

1.8 a. Find the following sums:

$$\sum_{i=0}^{\infty} \frac{1}{4^i} = ?$$

$$S = 1 + \frac{1}{4} + \frac{1}{16} + \dots$$

$$\frac{S}{4} = \frac{1}{4} + \frac{1}{16} + \dots$$

$$S - \frac{S}{4} = 1$$

$$S = \frac{4}{3}$$

1.8 b.

$$\sum_{i=0}^{\infty} \frac{i}{4^i} = ?$$

$$S = 0 + \frac{1}{4} + \frac{2}{16} + \frac{3}{64} + \dots$$

$$4S = 1 + \frac{2}{4} + \frac{3}{16} + \dots$$

$$4S - S = 1 + \frac{1}{4} + \frac{1}{16} + \dots$$

$$4S - S = \frac{4}{3} \text{ (from previous exercise)}$$

$$S = \frac{4}{9}$$

2.1 Sort the following according to their running time in ascending order:

$N, \sqrt{N}, N\sqrt{N}, N^2, N\log N, N\log(\log(N)), N\log N\log N, N\log(N^2), \frac{2}{N}, 2^N, 2^{N/2}, 37, N^2\log N, N^3$

In sorted order:

$\frac{2}{N}, 37, \sqrt{N}, N, N\log(\log(N)), N\log N, N\log(N^2), N\log N\log N, N\sqrt{N}, N^2, N^2\log N, N^3, 2^{N/2}, 2^N$

2.7 For the following loops provide the final value of sum and the running time.

```
int sum = 0;
for(int i=1; i <= n; i++)
    sum = sum + 1;
```

$$\sum_{i=1}^n 1 = 1 + 1 + 1 + \dots = n \quad \text{running time} = O(n)$$

```
int sum = 0;
for(int i=1; i <= n; i++)
    for(int j=1; j<=i; j++)
        sum = sum + 1;
```

$$\sum_{i=1}^n \sum_{j=1}^i 1 = \sum_{i=1}^n i = \frac{n(n+1)}{2} \quad \text{running time} = O(n^2)$$

2.11 An algorithm runs for 0.5ms for an input size of 100, what will be the running time for input size 500 if its time complexity is: a) linear, b) logarithmic, c) quadratic?

Linear is: $T(n) = n$

$$\frac{T(n_1)}{T(n_2)} = \frac{n_1}{n_2} = \frac{t_1}{t_2}$$

$$\frac{100}{500} = \frac{0.5}{t_2} \text{ (solve for } t_2 \text{)}$$

Logarithmic is $T(n) = \log(n)$

$$\frac{T(n_1)}{T(n_2)} = \frac{\log(n_1)}{\log(n_2)} = \frac{t_1}{t_2}$$

$$\frac{\log(100)}{\log(500)} = \frac{0.5}{t_2} \text{ (solve for } t_2 \text{)}$$

Quadratic is $T(n) = n^2$

$$\frac{T(n_1)}{T(n_2)} = \frac{n_1^2}{n_2^2} = \frac{t_1}{t_2}$$

$$\frac{100^2}{500^2} = \frac{0.5}{t_2} \text{ (solve for } t_2 \text{)}$$

2.12 An algorithm processes 100 inputs in 0.5 ms. How many inputs it will be able to process in 1 minute if its time complexity is a) linear, b) logarithmic, c) quadratic?

1 minute = 60x1000 = 60000ms

Linear is: $T(n) = n$

$$\frac{T(n_1)}{T(n_2)} = \frac{n_1}{n_2} = \frac{t_1}{t_2}$$

$$\frac{100}{n_2} = \frac{0.5}{60000} \text{ (solve for } n_2 \text{)}$$

Logarithmic is: $T(n) = \log(n)$

$$\frac{T(n_1)}{T(n_2)} = \frac{\log(n_1)}{\log(n_2)} = \frac{t_1}{t_2}$$

$$\frac{\log(100)}{\log(n_2)} = \frac{0.5}{60000} \text{ (solve for } n_2 \text{)}$$

Quadratic is $T(n) = n^2$

$$\frac{T(n_1)}{T(n_2)} = \frac{n_1^2}{n_2^2} = \frac{t_1}{t_2}$$

$$\frac{100^2}{n_2^2} = \frac{0.5}{60000} \text{ (solve for } n_2\text{)}$$

2.15 Give an efficient algorithm to determine if there exists an integer i such that $A_i = i$ in an array of integers $A_1 < A_2 < A_3 < \dots < A_N$. What is the running time of your algorithm?

Let c be an integer between 1 and N . Then,

-if $A_c < c$, then $A_i < i$, for $i = 1$ to $(c-1)$

Proof by induction:

Base case: $A_c < c$

Inductive step: $A_c < c$ implies $A_{c-1} < c-1$

Since $A_{c-1} < A_c < c$ is given, A_c can be at most $c-1$ giving $A_{c-1} < c-1$

Through induction (as a chain of implies) we can prove up to $i=1$

-if $A_c > c$, then $A_i > i$, for $i = (c+1)$ to N

Proof is similar

Algorithm:

Check if the middle element satisfies $A_i = i$, and if it does the answer is yes. If $A_i < i$ we can apply the same strategy to the subarray to the right of the middle element. If $A_i > i$ we can apply the same strategy to the subarray to the left of the middle element.

Finally, this boils down to binary search, thus algorithm is $O(\log N)$