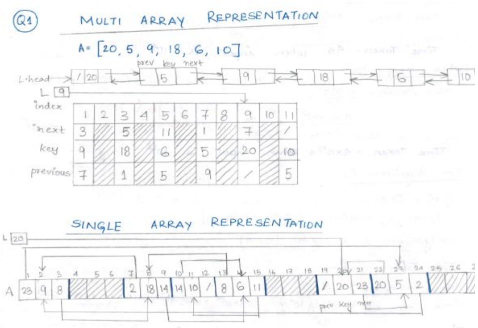
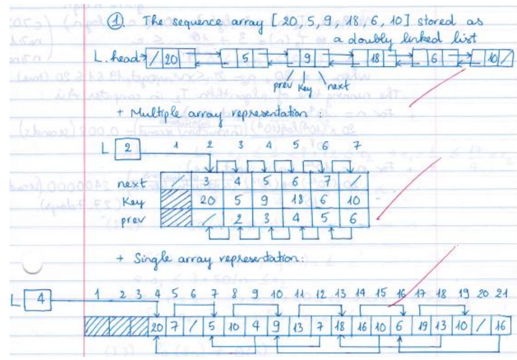
1. (10%) Draw a diagram of the multiple-array representation for the sequence array [20, 5, 9, 18, 6, 10] stored as a doubly linked list. Also draw a diagram of the single-array representation for the same sequence array.

Example 1: (From Parnika)



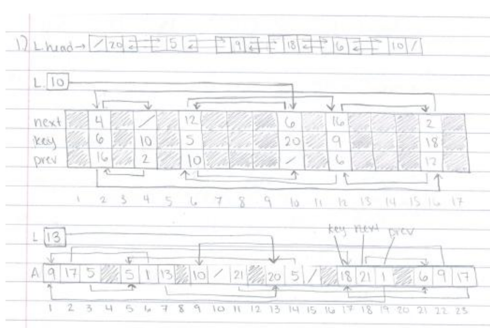
H1.1

Example 2: (from Nhi)



H1.2

Example 3: (from Kate)



H1.3

2. (10%) Assume that algorithm A_1 takes time roughly $T_1(n) = 3n^2 + 2n + 6$ and algorithm A_2 takes time roughly $T_2(n) = 3n\ lg(n) + 10$, and suppose that computer A's CPU runs 10^8 instructions/sec. When the input size equals to 10^4 and 10^{12} respectively, how long will algorithm A_1 take to finish for each input size? How long will algorithm A_2 take to finish for each input size? (Hint: Big O)

Answer:

Algorithm $T_1(n) = 3n^2 + 2n + 6$ ➔ The complexity $= O(n^2)$

$T_1(n) = 3n^2 + 2n + 6 \le c*n^2$ and find the value c and n_0

→ $3 + 2/n + 6/n^2 \le c$ → when c=4, and $n_0 = 4 \le n$,

The run time of Algorithm T_1 in computer A:

(1) $n = 10^4$: $4*(10^4)^2/10^8$ ins/sec = 4 sec

(2) $n = 10^{12}$: $4*(10^{12})^2/10^8$ ins/sec = $4*10^{16}$ sec ($1.27*10^9$ years)

$T_2(n) = 3n\ lg(n) + 10$ ➔ The complexity $= O(n\ lg(n))$

$3n\ lg(n) + 10 \le c * n\ lg(n)$ → when c=4, and $n_0 = 5 \le n$, *$lg(5) = 2.322$*

➔ *$3n\ lg(5) + 10 = 34.829 \le 4 * 5\ lg(5)$* = 46.44

(1) $n = 10^4$: $4*(10^4)lg10^4/10^8$ ins/sec = $4*(10^4)*13.288/10^8$ = 0.0053 sec

(2) $n = 10^{12}$: $4*(10^{12})\ lg10^{12}/10^8$ ins/sec = $4*(10^{12})*39.86/10^8$

➔ $1.59*10^6$ sec (18.4 Days)

H1.4

3. (32%) For each of the following "=", identify the corresponding constants C1, C2, n_0 as appropriate. For each of "≠", lines, show they cannot possibly exist.

(3.1) $\frac{n^2}{4} - 8n + 12 = \Theta(n^2)$ = → g(n) = n^2

→ $c_1 n^2 \le n^2/4 - 8n + 12 \le c_2 n^2$

→ $c_1 \le 1/4 - 8/n + 12/n^2 \le c_2$

→ when n=1 : $c_1 \le 1/4 - 8 + 12 \le c_2$ → $c_1 = 1/4 \le 17/4 = c_2$

→ pick $c_1 = 1/4$, $c_2 = 17/4$ and $n_0 = 1 \le n$

(3.2) $3 + 50/n = \Theta(1)$ → g(n) = 1

→ $c_1 \le 3 + 50/n \le c_2$ → pick $c_1 = 3$, $c_2 = 53$ and $n_0 = 1 \le n$

(3.3) $\lg(2n) \ne \Theta(n)$ → g(n) = n

→ $c_1 n \le \lg(2n) \le c_2 n$ → even we can find c_2 and n_0 to satisfy $\lg(2n) \le c_2 n$ for O(n), however , since n grows faster than lg(2n), we cannot find any c_1 and n_0 to satisfy $c_1 n \le \lg(2n)$ for **Ω(n) ,** therefore $\lg(n) \ne \Theta(n)$.

(3.4) $n^{100} = O(2^n)$ → $g(n)=2^n$

→ $n^{100} \le c2^n$ → pick c = 1 and $n_0 = 2^{10} \le n$

$(2^{10})^{100} \le 1 * 2^{2^{10}}$ → $2^{1000} \le 2^{1024}$

……….

$(2^{11})^{100} \le 1 * 2^{2^{11}}$ → $2^{1100} \le 2^{2048}$

………

……..

(3.5) $10n-7 = O(n^2)$ → $g(n) = n^2$

→ $10n-7 \le cn^2$

→ $(10n-7)/n^2 \le c$ pick c =4, $n_0 = 1 \le n$

(3.6) $4n^{1/2} \ne o(n^{1/2})$ (note the little-o)

→ $g(n) = n^{1/2}$ → By the definition $o(g(n)) = \{ f(n) \mid \forall c, \exists n_0 \forall n > n_0, 0 \le f(n) \le cg(n)\}$

→ $\lim_{n\to\infty} \frac{4n^{1/2}}{n^{1/2}} = 4$ → $4n^{1/2} \ne o(n^{1/2})$

We cannot find all constant values c and n_0 to satisfy $0 \le 4n^{1/2} \le c\, n^{1/2}$

$f(n) = o(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

(3.7) $\frac{n^3}{3} \ne \omega(n^3)$ → $g(n) = n^3$

By the definition : $\omega(g(n)) = \{ f(n) \mid \forall c, \exists n_0 \forall n > n_0, 0 \le cg(n) \le f(n)\}$

$f(n) = \omega(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

→ $\lim_{n\to\infty} \frac{n^3/3}{n^3} = 1/3 \ne \infty$

We cannot find all const values c and n_0 to satisfy $0 \le c\, n^3 \le n^3 / 3$

(3.8) $123n + 321 = \Theta(n)$ → g(n) = n

→ $c_1 n \le 123n + 321 \le c_2 n$

→ $c_1 \le 123 + 321/n \le c_2$

→ pick $c_1 = 123$ and $c_2 = 444$, $n_0 = 1 \le n$

4.(10%) Answer Problem 3-1 (asymptotic behavior of polynomials) on page 61 of textbook.

Let

$$p(n) = \sum_{i=0}^{d} a_i n^i ,$$

where $a_d > 0$, be a degree-d polynomial in n, and let k be a constant. Use the definitions of the asymptotic notations to prove the following properties.

a. If $k \ge d$, then $p(n) = O(n^k)$.

b. If $k \le d$, then $p(n) = \Omega(n^k)$.

c. If $k = d$, then $p(n) = \Theta(n^k)$.

d. If $k > d$, then $p(n) = o(n^k)$.

e. If $k < d$, then $p(n) = \omega(n^k)$.

4.a If $k \geq d$, then $p(n) = O(n^k)$.

If we pick any $c > 0$, then, the end behavior of $cn^k - p(n)$ is going to infinity, in particular, there is an n_0 so that for every $n \geq n_0$, it is positive, so, we can add $p(n)$ to both sides to get $p(n) < cn^k$.

4.b If $k \leq d$, then $p(n) = \Omega(n^k)$.

If we pick any $c > 0$, then, the end behavior of $p(n) - cn^k$ is going to infinity, in particular, there is an n_0 so that for every $n \geq n_0$, it is positive, so, we can add cn^k to both sides to get $p(n) > cn^k$.

4.c If $k = d$, then $p(n) = \Theta(n^k)$.

We have by the previous parts that $p(n) = O(n^k)$ and $p(n) = \Omega(n^k)$. So, by Theorem 3.1, we have that $p(n) = \Theta(n^k)$.

4.d If $k > d$, then $p(n) = o(n^k)$.

$$\lim_{n\to\infty} \frac{p(n)}{n^k} = \lim_{n\to\infty} \frac{n^d(a_d + o(1))}{n^k} < \lim_{n\to\infty} \frac{2a_d n^d}{n^k} = 2a_d \lim_{n\to\infty} n^{d-k} = 0$$

4.e If $k < d$, then $p(n) = \omega(n^k)$.

$$\lim_{n\to\infty} \frac{n^k}{p(n)} = \lim_{n\to\infty} \frac{n^k}{n^d O(1)} < \lim_{n\to\infty} \frac{n^k}{n^d} = \lim_{n\to\infty} n^{k-d} = 0$$

5. (18%) Identify and EXPLAIN all elements *F* of the set { O, Ω, Θ,ω, o } such that the f(n) = *F*(g(n)) for each of the following asymptotic relations. Thus that if f(n)= O(g(n)) and f(n)= Ω (g(n)) and f(n) = Θ(g(n)) are the only three valid asymptotic relationships between f(n) and g(n) , answer O, Ω, Θ.

(5.1) f(n) = $n^2 + 5$, $g(n) = 3n^2 + 4n$

→ f(n) = O(g(n) , we can find c >0 and $n_0 \leq$ **n** to satisfy $n^2 + 5 \leq c\,(3n^2 + 4n)$,
Pick c = 1, $n_0 = 1 \leq$ **n**

→ f(n) ≠ Ω(g(n) , we cannot find c >0 and $n_0 \leq$ **n** to satisfy $c\,(3n^2 + 4n) \leq n^2 + 5$

→ f(n) ≠ Θ(g(n)), since f(n) ≠ Ω(g(n)

→ f(n) ≠ ω (g(n), by the definition, $f(n) = \omega(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

$$\lim_{n\to\infty} \frac{n^2+5}{3n^2+4n} = 1/3 \qquad f(n) = o(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 1/3$$

→ f(n) ≠ o(g(n)) ,

→ F = {O }

(5.2) f(n) = $2*\log_2 n$, $g(n) = \log_3(2n)$

→ f(n) = O(g(n) , we can find c >0 and $n_0 \leq$ **n** to satisfy $2*\log_2 n \leq c\,(\log_3(2n))$
Pick c = 2, $n_0 = 1 \leq$ **n**

→ f(n) = Ω(g(n) , we can find c >0 and $n_0 \leq$ **n** to satisfy $c\,(\log_3(2n)) \leq 2*\log_2 n$
Pick c = lg3 =1.58, $n_0 = 2 \leq$ **n**

→ f(n) = Θ(g(n)), since f(n) = O(g(n) and f(n) = Ω (g(n)

→ f(n) ≠ ω (g(n), by the definition, $f(n) = \omega(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

$$\lim_{n\to\infty} \frac{2\log_2 n}{\log_3(2n)} = \frac{2\log_3 n}{\log_3^{\ 2}} \Big/ \log_3(2n) = \frac{2\log_3 n}{\log_3^{\ 2}[\log_3 2 + \log_3 n\ ]} = \frac{2}{\log_3^{\ 2}} \neq \infty$$

→ f(n) ≠ o(g(n)) , $f(n) = o(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

→ F = {O, Ω ,Θ}

(5.3) $f(n) = 2^n + 3n$, $g(n) = 3^n + 2n + 1$

→ $f(n) = O(g(n))$, we can find $c > 0$ and $n_0 \le$ **n** to satisfy $2^n + 3n \le c\,(3^n + 2n + 1)$, Pick $c = 1$, $n_0 = 1 \le$ **n**

→ $F(n) \ne \Omega(g(n))$, we cannot find $c > 0$ and $n_0 \le$ **n** to satisfy $c\,(3^n + 2n + 1) \le 2^n + 3n$

→ $f(n) \ne \Theta(g(n))$, since $f(n) \ne \Omega(g(n))$

→ $f(n) \ne \omega\,(g(n))$, by the definition, $f(n) = \omega(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

$$\lim_{n\to\infty} \frac{2^n + 3n}{3^n + 2n + 1} = 0$$

→ $f(n) = o(g(n))$, $f(n) = o(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

→ F = {O, o }

(5.4) $f(n) = n^{1/2}$, $g(n) = 4*\log n$

→ $f(n) \ne O(g(n))$, we cannot find $c > 0$ and $n_0 \le$ **n** to satisfy $n^{1/2} \le c\,(4*\log n)$

→ $F(n) = \Omega(g(n))$, we can find $c > 0$ and $n_0 \le$ **n** to satisfy $c\,(4*\log n) \le n^{1/2}$

Pick $c = 1$, $n_0 = 1 \le$ **n**

→ $f(n) \ne \Theta(g(n))$, since $f(n) \ne O(g(n))$

→ $f(n) = \omega\,(g(n))$, by the definition, $f(n) = \omega(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

$$\lim_{n\to\infty} \frac{\sqrt{n}}{4\log n} = \infty$$

→ $f(n) \ne o(g(n))$, $f(n) = o(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

→ F = {Ω, ω}

(5.5) $f(n) = n^2 + 1$, $g(n) = 3n - 2$

→ $f(n) \ne O(g(n))$, we cannot find $c > 0$ and $n_0 \le$ **n** to satisfy $n^2 + 1 \le c\,(3n - 2)$

→ $F(n) = \Omega(g(n))$, we can find $c > 0$ and $n_0 \le$ **n** to satisfy $c\,(3n - 2) \le n^2 + 1$, $c=1$, $n_0 = 1 \le$ **n**

→ $f(n) \ne \Theta(g(n))$, since $f(n) \ne O(g(n))$

→ $f(n) = \omega\,(g(n))$, by the definition, $f(n) = \omega(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

$$\lim_{n\to\infty} \frac{n^2 + 1}{3n - 1} = \infty$$

→ $f(n) \ne o(g(n))$, since $f(n) = \omega\,(g(n))$ and $f(n) = o(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

→ F = {Ω, ω }

(5.6) $f(n) = 2n + 1$, $g(n) = 3*\log^2 n + 2$

→ $f(n) \ne O(g(n))$, we cannot find $c > 0$ and $n_0 \le$ **n** to satisfy $2n + 1 \le c\,(3*\log^2 n + 2)$

→ $F(n) = \Omega(g(n))$, we can find $c > 0$ and $n_0 \le$ **n** to satisfy $c\,(3*\log^2 n + 2) \le 2n + 1$

Pick $c = 1$, $n_0 = 1 \le$ **n**

→ $f(n) \ne \Theta(g(n))$, since $f(n) \ne O(g(n))$

→ $f(n) = \omega\,(g(n))$, by the definition, $f(n) = \omega(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

$$\lim_{n\to\infty} \frac{2n + 1}{3\log^2 n + 2} = \infty$$

→ $f(n) \ne o(g(n))$, $f(n) = o(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

→ F = {Ω, ω}

6. (20%) Convert the pseudo codes of the INSERTION-SORT(A) algorithm on page 18 and pseudo codes of the MERGE-SORT(A,p,r) algorithm on page 34 of the textbook respectively into executable Java codes to sort array A[18, 25, 6, 9, 15, 12, 5, 20, 11, 30]. Show all your source codes for both algorithms and take screen shots of **each intermediate step of sorting results**. (Note: 0 point will be received if you did not use the pseudo codes from the textbook.)