

แบบฝึกหัด บทที่ 1 การวิเคราะห์เวลา

1. จงประมาณเวลาโดยวิธีนับรอบของ basic operation โดยเขียนในรูปของ $T(n)$

1.1	<pre>sum = 0; for(i=2; i<n+1; i++) sum++;</pre>
1.2	<pre>sum=0; for(i=0; i<n; i=i+2) sum++;</pre>
1.3	<pre>sum=0; for(i=1;i<n; i=i*2) sum++;</pre>
1.4	<pre>sum=0; for(i=1;i<n/2; i++) sum++;</pre>
1.5	<pre>sum=0; for(i=0;i<n; i++) for(j=0;j<n*n; j++) sum++;</pre>
1.6	<pre>sum=0; for(i=0; i<n; i++) for(j=0; j<i; j++) sum++;</pre>
1.7	<pre>sum=0; for(i=0; i<n; i++) for(j=0; j<i*i; j++) for(k=0; k<j; k++) sum++;</pre>
1.8	<pre>sum=0; for(i=0; i<n; i++) sum++; val =1; for(j=0; j<n*n; j++) val = val + n;</pre>
1.9	<pre>count=1; while(n > 1) { count += 1; n = n/3; }</pre>
1.10	<pre>i = 0; j = n; while(i<j) { i=i+3; j=j+5; }</pre>

แบบฝึกหัด บทที่ 1 การวิเคราะห์เวลา

1.11	<pre>sum = 0; for(i=0;i<n;i=i+2) sum++; for(i=0;i<n/2;i=i+5) sum++;</pre>
------	---

2. กำหนดให้ n เป็นขนาดของอินพุต จงเรียงลำดับฟังก์ชันต่อไปนี้จากน้อยไปมาก

$5n^3 + 2n^2 + 30$ $7\log n$ $10n$ n^{100} 5,000,000 $0.5n^3$ $n^{1/3}$ $100*2^n$

3. จากอัลกอริทึมต่อไปนี้ จงวิเคราะห์เวลาทำงานของอัลกอริทึมโดยใช้กรณีเลวร้าย (worst case) กรณีดีสุด (Best case) และกรณีเฉลี่ย (Average case)

3.1	<pre>function reverseArray(arr[], start, end) while start < end swap arr[start] and arr[end] start = start + 1 end = end - 1 end while</pre>
-----	---

3.2	<pre>function bubbleSort(A) n = length(A) for i from 1 to n - 1 for j from 0 to n - i - 1 if A[j] > A[j + 1] swap A[j] and A[j + 1]</pre>
-----	--

3.3	<pre>function BinarySearch (A[], l, r, k) while l <= r : m = l + (r - l) / 2 if (A[m] == k) return m if (A[m] < k) l = m + 1 else r = m - 1 end while return -1</pre>
-----	---

4. จากฟังก์ชัน Big-Oh ในแต่ละข้อ จงเขียนให้อยู่ในรูปของฟังก์ชันบิกโอด้วยง่าย และใกล้เคียงที่สุด

ข้อ	ฟังก์ชัน Big-Oh	คู่ตอบ
4.1	$O(3n - 5)$	$O(n)$
4.2	$O(2n^3 + 4n)$	

แบบฝึกหัด บทที่ 1 การวิเคราะห์เวลา

4.3	$O(1/n + 1)$	
4.4	$O(7n^8 + 3n^2 - 4)$	
4.5	$O(n + \log n + 8000)$	

5. กำหนดให้ $T(n) = 2n^3 + 4n$ ข้อใดกล่าวถูกต้อง

- (a) $T(n) = O(n^3)$
- (b) $T(n) = O(n^5)$
- (c) $T(n) = O(n^3 + 5n^2)$
- (d) $T(n) = O(n)$
- (e) $T(n) = \Omega(n^3)$
- (f) $T(n) = \Omega(n^2)$
- (g) $T(n) = \Theta(n^3)$
- (h) $T(n) = \Theta(n^4)$

6. จงพิสูจน์แต่ละข้อว่าเป็นจริงหรือเท็จ

- 6.1 $T(n) = 3n^3 + 2n + 7 \in \Theta(n^3)$
- 6.2 $T(n) = n^3 + 20n + 1 \in \Omega(n^2)$
- 6.3 $T(n) = n^3 + 20n + 1 \in O(n^2)$

7. จงแก้สมการ recurrence

- 7.1 $X(n) = X(n-1) + 5$ for $n > 1$, $X(1) = 0$
- 7.2 $X(n) = 3X(n-1)$ for $n > 1$, $X(1) = 4$
- 7.3 $X(n) = X(n/2) + n$ for $n > 1$, $X(1) = 1$

8. วิเคราะห์ Big-oh ของอัลกอริทึม ต่อไปนี้

8.1	<pre> Algorithm Parallel-Product(A[1..n]); if n = 1 then return; for i := 1 to n/2 do A[i] := A[i]*A[i+n/2]; call Parallel-Product(A[1..n/2]); </pre>
8.2	<pre> Algorithm Q(n) //Input: A positive integer n if n = 1 return 1 else return Q(n - 1) + 2 * n - 1 </pre>

แบบฝึกหัด บทที่ 1 การวิเคราะห์เวลา

8.3 long power(long x, long n)
 if (n == 0)
 return 1;
 else
 return x * power(x, n-1);

8.4 long power(long x, long n)
 if (n==0) return 1;
 if (n==1) return x;
 if ((n % 2) == 0)
 return power(x*x, n/2);
 else
 return power(x*x, n/2) * x;