TDA416, LAB1

Jesper Jaxing och Oskar Henriksson

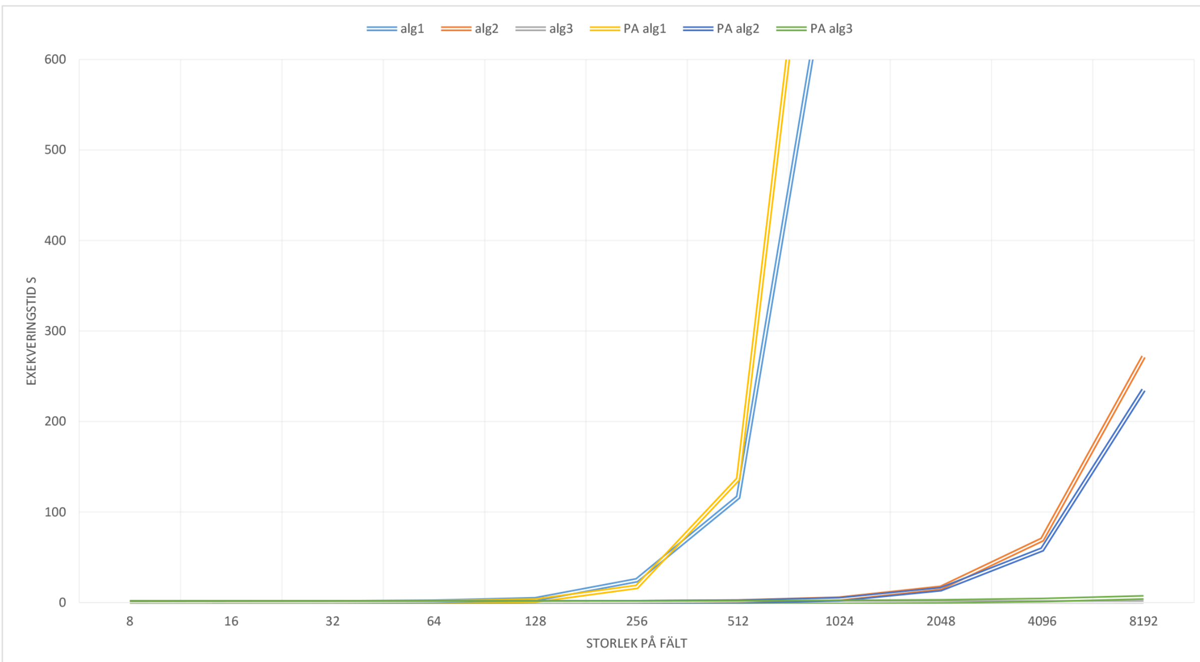
January 2016

1 LAB1 del2

En feltolkning av oss gjorde att tre pedantiska analyser togs fram medan bara en matematisk uppskattning. Eftersom de pedantiska är mer omfattande ser vi inte det som ett problem.

1.1 Illustration

I grafen står PA för pedantisk analys och alg, för algoritm/metod.



1.2 Metod 1

1.2.1 Pedantisk analys

$+1$	int maxSum = 0;
$+1+\sum_{i=0}^{n-1}(2+'innehåll')+1$	for(int i=0; i < a.length; i++)
$+1+\sum_{j=i}^{n-1}(2+'innehåll')+1$	for(int j=i; j < a.length; j++)
$+1$	int thisSum = 0;
$+1+\sum_{k=i}^{j}(2+'innehåll')+1$	for(int k=i; k <= j; k++)
$+1$	thisSum += a[k];
$+1$	'sista' k <= j
$+1$	if (thisSum > maxSum)
$+1$	maxSum = thisSum;
$+1$	seqStart = i;
$+1$	seqEnd = j;
$+1$	return maxSum;

Tabellen ger följande:

$$4+\sum_{i=0}^{n-1}(4+\sum_{j=i}^{n-1}(8+\sum_{k=i}^{j}(3))) \tag{1}$$

som konvergerar till:

$$4+\frac{n(n^2+8n+15)}{2} \in O(n^3) \tag{2}$$

1.2.2 Handviftning

Man kan uppskatta denna algoritm till $O(n^3)$ efter som den innehåller tre loop som beror på indatan (innersta loppen går till j, j beror på indatan).

1.3 Metod 2

1.3.1 Pedantisk analys

$+1$	int maxSum = 0;
$+1+\sum_{i=0}^{n-1}(2+'innehåll')+1$	for(int i = 0;i < a.length; i++);
$+1$	int thisSum = 0;
$+1+\sum_{j=i}^{n-1}(2+'innehåll')+1$	for(int j = i ;i < a.length;i++)
$+1$	thisSum += a[j];
$+1$	if(thisSum > maxSum)
$+1$	maxSum = thisSum;
$+1$	seqStart = i;
$+1$	seqEnd = j;
$+1$	return maxSum;

Tabellen ger följande:

$$4+\sum_{i=0}^{n-1}(3+\sum_{j=i}^{n-1}(7)) \tag{3}$$

som konvergerar till

$$\frac{n(7n+13)}{2}+4 \in O(n^2) \tag{4}$$

1.3.2 Handviftning

Man kan uppskatta denna algoritm till $O(n^2)$ efter som den innehåller två loop som beror på indatan.

1.4 Metod 3

1.4.1 Pedantisk analys

$+1$	int maxSum = 0;	
$+1$	int thisSum = 0;	
$+1\ +1+\sum_{j=0}^{n-1}(2+'innehåll')+1$	for(int i = 0, j = 0;i < a.length; j++);	
$+1$	thisSum += a[j];	
$+1$	if(thisSum > maxSum)	'jobbigaste' if-satsen
$+1$	maxSum = thisSum;	
$+1$	seqStart = i;	
$+1$	seqEnd = j;	
$+1$	return maxSum;	

Tabellen ger följande:

$$6+\sum_{j=0}^{n-1}(7) \tag{5}$$

som konvergerar till

$$6+7n \in O(n) \tag{6}$$

1.4.2 Handviftning

Man kan uppskatta denna algoritm till O(n) efter som den endast innehåller en loop som beror på indatan.

1.4.3 Matematisk uppskattning

$+\sum_{j=0}^{n-1}('innehåll')$	for(int i = 0, j = 0;i < a.length; j++);

Tabellen ger följande:

$$\sum_{j=0}^{n-1}(1) \tag{7}$$

som konvergerar till

$$n \in O(n) \tag{8}$$