Fakulteti i Shkencës Kompjuterike

Algoritmet dhe strukturat e të dhënave

Msc. Menduh Çerkezi

Analiza e algoritmeve

Lidhur me analizën e algoritmeve njihemi me nocionet si “kompleksiteti kohor” (Ang: time complexity) dhe “kompleksiteti hapësinor” (Ang: space complexity).

Kompleksiteti kohor – është numri i hapave që duhet të ekzekutohen si funksion i përmasave të hyrjeve. d.m.th. koha e ekzekutimit varret nga hyrja.

Kompleksiteti hapësinor – është madhësia e memories e cila përdoret si funksion i përmasave të hyrjeve.

Kurse koha ekzekutimit (Ang: running time) është numri i instruksioneve që bëhen për tu ekzekutuar programi.

Marrim një shembull fillestar ku si hyrje do të jetë vektori *A* me *n* elemente. Dhe dëshirojmë t’i shtypim të gjitha vlerat. Ta kemi parasysh që këtu vektori fillon prej indeksit 1. Pseudokodi vijon:

```
for i:=1 to length(A)
 print A[i]
```

Pra do të bëhen *n* krahasime për të shtypur të gjithë elementet e matricës, si dhe një krahasim më shumë në unazën e fundit, atëherë kur $i=n+1$ d.m.th gjithsejt n+1, **ku ne do ta shënojmë si *O(n)***, pasi që numri 1 është pothuajse i parëndësishëm kur kemi të bëjmë me numra të mëdhenj. **Pra e tëra që na intereson është numri i krahasimeve për vlera të mëdha të n. Këtu e pamë se koha e ekzekutimit të algoritmit është n çka do të thotë se rritja e saj do të jetë lineare. Me anë të kalkulimit të kohës së ekzekutimit caktojmë rritjen e funksionit të një algoritmi.**

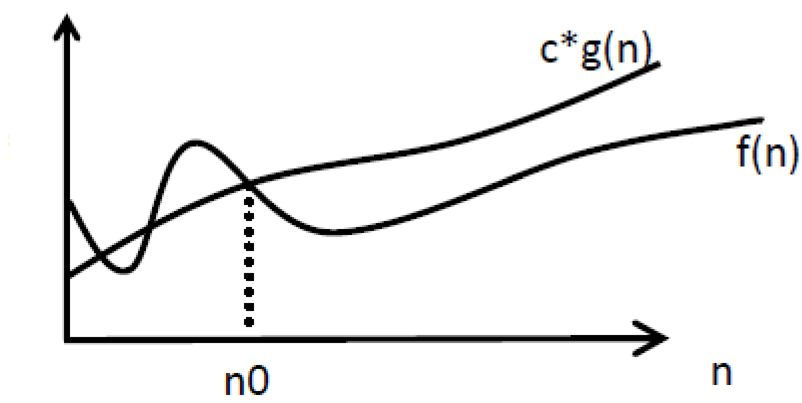
Rritja e funksioneve

Zakonisht me rritjen e *n* rritet edhe kohëzgjatja e ekzekutimit të programit. Varësisht nga hyrjet, i njëjti program mund të ekzekutohet për një kohë të shkurtër apo të gjatë. **Psh.** n**ë të shumtën e rasteve neve na intereson se si sillet algoritmi në skenarin më të pa favorshëm (Ang: worst case scenario)**.
Qe t’i krahasojmë dy algoritme, ne duhet ta dimë se sa shpejtë rriten ato algoritme. Për ketë arsye njihemi me simbolin e madh O (Ang: Big-O notation) dhe me simbolin e madh Omega (ang. Big-Omega notation).

Simboli i madh O (ang. Big O notation)

Definicion:

```
Le të jenë f dhe g  funksione  prej bashkësisë së numrave të plotë apo
real në bashkësinë e numrave real. Ne themi që f(n) = O(g(n)) nëse
ekzistojnë konstantet C dhe n0 ashtu që
|f(x)| <= C|g(x)| , ku n > n0
dhe lexohet f(n) është O(g(n))
```



Do të thotë që nëse gjejmë një pikë C, atëherë algoritmi me kompleksitetin kohor g(n) do të jetë gjithmonë kufiri i lartë i algoritmit f(n).

Shembull:

Vërtetoni se $f(n) = n^2 + 2n + 1$ është $O(n^2)$

Në shembullin e mësipërm nevojitet të gjendet se për çdo $n>1$, ekziston një konstantë C ashtu që $|f(n)| <= C|g(n)|$.

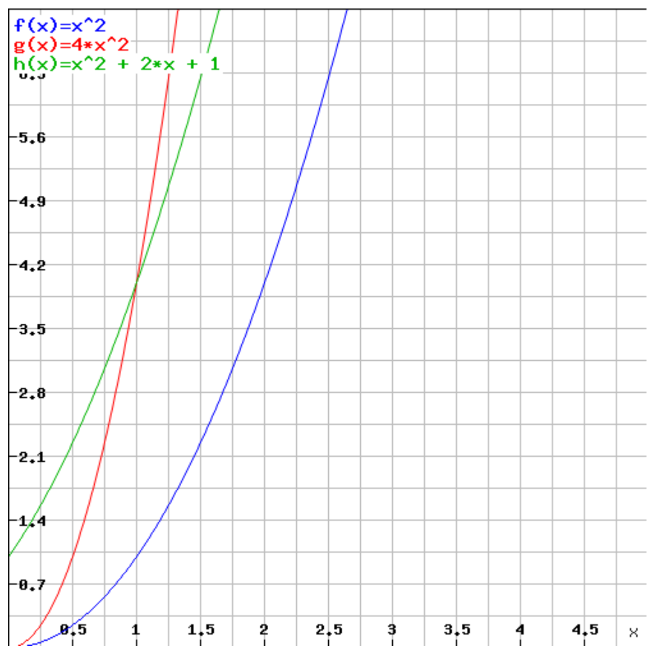
Zgjidhja:

$n^2 + 2n + 1$ e dimë që është më e vogël se $n^2 + 2n^2 + n^2 = 4n^2$

dmth $4n^2 <= Cn^2$. Kjo është e saktë sa herë qe $C=4$ dhe $n>=1$

dmth $n^2 + 2n + 1$ është $O(n^2)$

n	$n^2 + 2n + 1$	n^2	$4n^2$
0	1	0	0
1	4	1	4
2	9	4	16
3	16	9	36
4	25	16	64
...	...	...	...



Shembull:

Vërtetoni se f(n) = 3n + 7 është O(n)

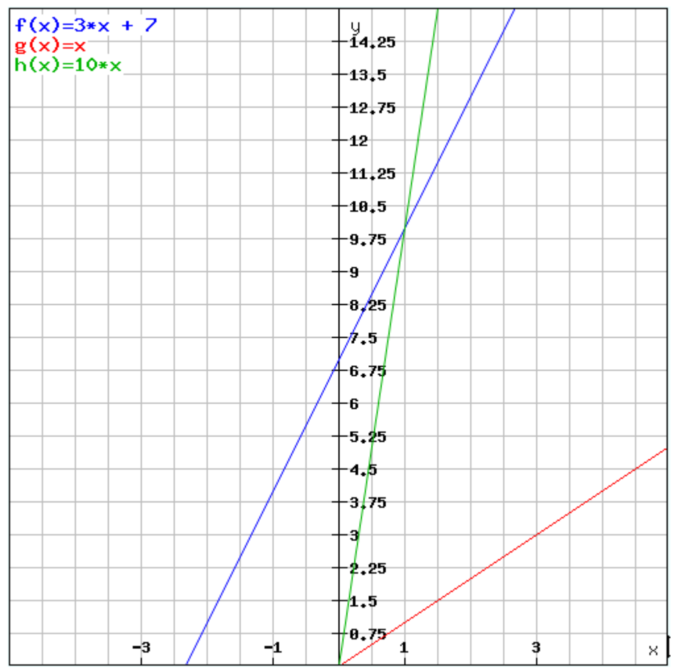
Zgjidhja:

3n + 7 e dimë që është më e vogël se 3n + 7n = 10n

dmth 10n <= Cn. Kjo është e saktë sa herë qe C=10 dhe n>=1

dmth 3n + 7 është O(n)

n	3n + 7	n	10n
0	7	0	0
1	10	1	10
2	13	2	20
3	16	3	30
4	19	4	40
…	…	…	…



Shembull:

Vërtetoni se $f(n) = (n + 1)^3$ është $O(n^3)$

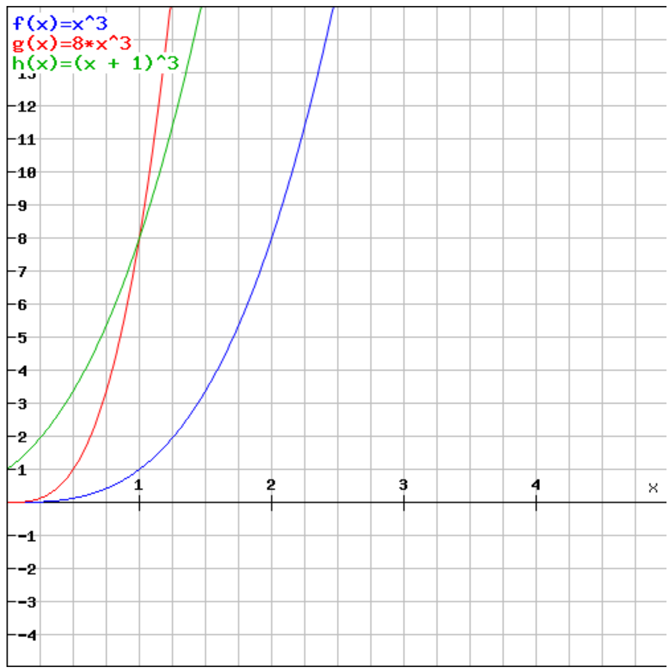
Zgjidhja:

$(n + 1)^3$ e dimë që është më e vogël se $(n + n)^3 = (2n)^3 = 8n^3$

dmth $8n^3 <= Cn^3$. Kjo është e saktë sa herë qe C=8 dhe n>=1

dmth $(n + 1)^3$ është $O(n^3)$

n	$(n + 1)^3$	n^3	$8n^3$
0	1	0	0
1	8	1	8
2	27	8	64
3	64	27	216
4	125	64	512
…	…	…	…



Shembull:

Vërtetoni se $f(n) = n^3 + 20n + 1$ është $O(n^3)$

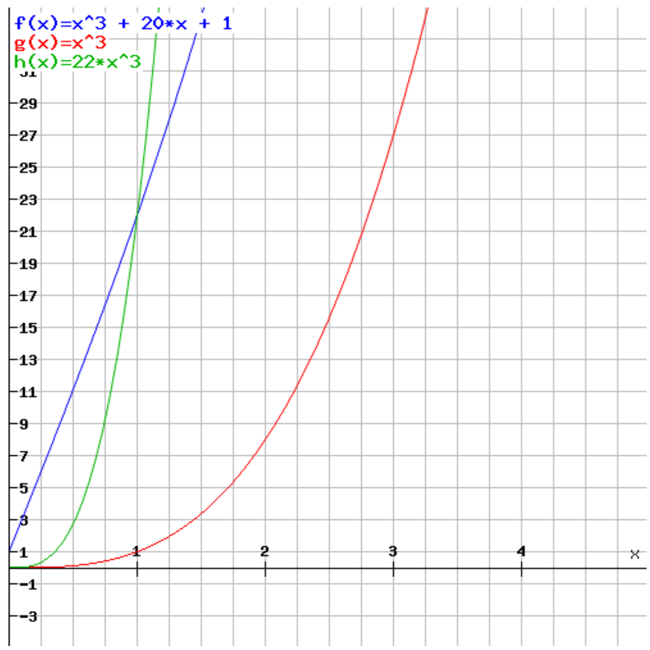
Zgjidhja:

$n^3 + 20n + 1$ e dimë që është më e vogël se $n^3 + 20n^3 + n^3 = 22n^3$

dmth $22n^3 <= Cn^3$. Kjo është e saktë sa herë qe C=22 dhe n>=1

dmth $n^3 + 20n + 1$ është $O(n^3)$

n	$n^3 + 20n + 1$	n^3	$22n^3$
0	1	0	0
1	22	1	22
2	49	8	176
3	88	27	594
4	145	64	1408
...	...	...	...



Shembull:

Vërtetoni se $f(n) = n^2 + 42n + 7$ është $O(n^2)$

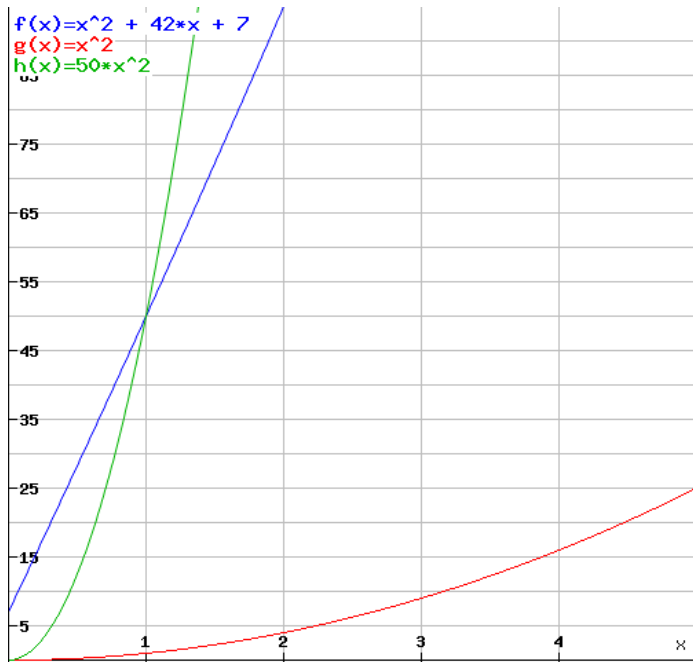
Zgjidhja:

$n^2 + 42n + 7$ e dimë që është më e vogël se $n^2 + 42n^2 + 7n^2 = 50n^2$

dmth $50n^2 <= Cn^3$. Kjo është e saktë sa herë qe C=50 dhe n>=1

dmth $n^2 + 42n + 7$ është $O(n^2)$

n	$n^2 + 42n + 7$	n^2	$50n^2$
0	7	0	0
1	50	1	50
2	95	4	200
3	143	9	450
4	191	16	800
…	…	…	…



Shembull:

Vërtetoni se $6n^4 - 2n^3 + 5$ është O(n^4)

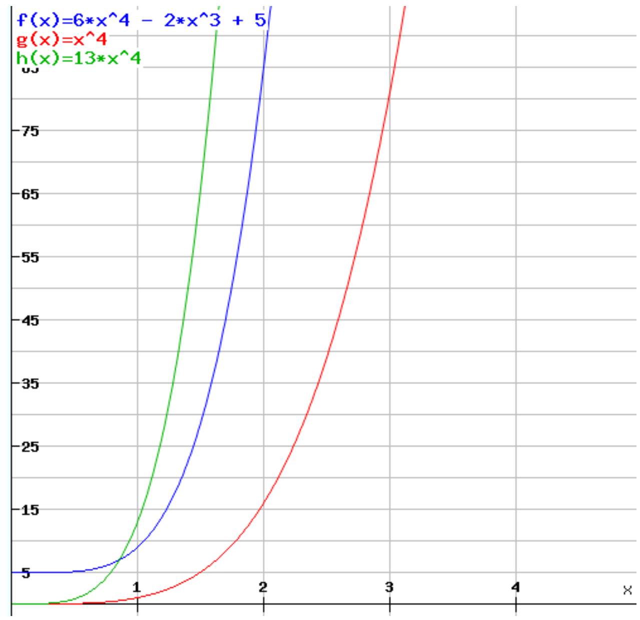
Zgjidhja:

$6n^4 - 2n^3 + 5$ e dimë që është më e vogël se $6n^4 + 2n^4 + 5n^4 = 13n^4$

dmth $13n^4 \leq Cn^4$. Kjo është e saktë sa herë qe C=13 dhe x ≥ ~1

dmth $6n^4 - 2n^3 + 5$ është O(n^4)

n	$6n^4 - 2n^3 + 5$	n^4	$13n^4$
0	5	0	0
1	9	1	13
2	85	16	208
3	437	81	1053
4	1413	256	3328
...	...	...	...

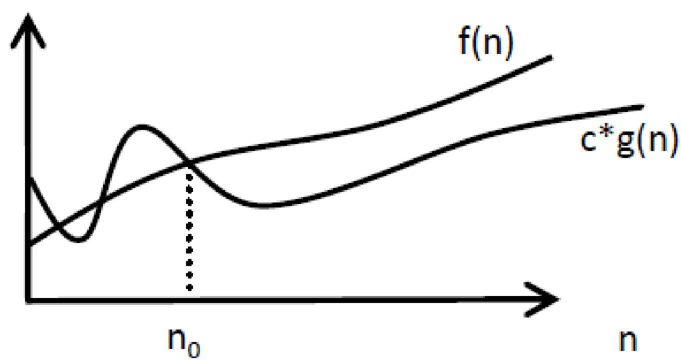


Simboli i madh Omega (Ω)

E pamë se simboli O na tregon ku gjendet kufiri i sipërm, mirëpo nuk na jep informatë lidhur me kufirin e poshtëm. Kjo do të thotë qe e dimë qe një algoritëm kufizohet me një lakore për së larti, që nuk do ta kalojë atë, mirëpo nuk e dimë se cila është lakorja e poshtme që do të na kishte dhënë informatë më të saktë se ku shtrihet algoritmi. Këtu vjen në shprehje simboli omega Ω që është i ngjashëm me O dhe përkufizohet kështu.

Definicion:

Le të jenë *f* dhe *g* funksione prej bashkësisë së numrave të plotë apo real në bashkësinë e numrave real. Ne themi që *f(n) = Ω (g(n))* nëse ekzistojnë konstantet pozitive *C* dhe *n0* ashtu që
| f(n) | >= C| g(n) | , ku n > n0
dhe lexohet f(n) është Ω (g(n))



Do të thotë që nëse gjejmë një pikë C, atëherë algoritmi me kompleksitetin kohor g(n) do të jetë gjithmonë kufiri i poshtë i algoritmit f(n).

Shembull:

Vërtetoni se $3n + 3$ është Ω(n)

Zgjidhja:

f(n) = 3n + 3; g(n) = n;

f(n) >= cg(n) --> Omega

Shihet qartë se 3n + 3 > 3n, për çdo n > 0.

Shembull:

Vërtetoni se $8n^3+5n^2+7$është $\Omega(n^3)$

Zgjidhja:

Duhet të vërtetojmë se $8n^3+5n^2+7 >= c\, n^3$

Shihet qartë se është e saktë për C = 1 dhe n >=1.

Shembull:

Vërtetoni se $5n^2 - 3n$ është $\Omega(n^2)$

Zgjidhja:

Duhet të vërtetojmë se $5n^2 - 3n >= c\, n^2$

Shihet qartë se është e saktë për C = 1 dhe n >=1.

n	$5n^2 - 3n$	n^2	$1n^2$
0	0	0	0
1	2	1	1
2	14	4	4
3	36	9	9
4	68	16	16
...	...	...	...

Shembull:

Vërtetoni se $n^3 + 20n$ është $\Omega(n^2)$

Zgjidhja:

Duhet të vërtetojmë se $n^3 + 20n >= c\, n^2$ (1)

Nëse e pjesëtojmë ekuacionin e mësipërm me n^2 atëherë kemi:

$$n + \frac{20}{n} \geq c$$

Shihet qartë se kushti për tu plotësuar (1) është për n $\geq n_0 = 5$ dhe c ≤ 9.

Pyetje ?

menduh.cerkezi@universitetiaab.com