

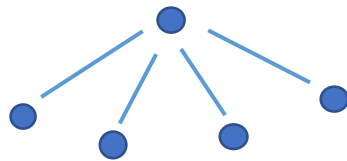
Progetto algoritmi

Il nostro primo algoritmo, avendo inizialmente inteso il nodo medio semplicemente come il nodo che aveva distanza uguale ad ogni coppia di nodi del grafo ma senza essere necessariamente parte del cammino minimo tra questi è stato sviluppato in questo modo:

L'algoritmo comincia a costruire un albero BFS avente il nodo da studiare come radice, quindi ad ogni livello calcola per quante coppie è medio in questo modo:

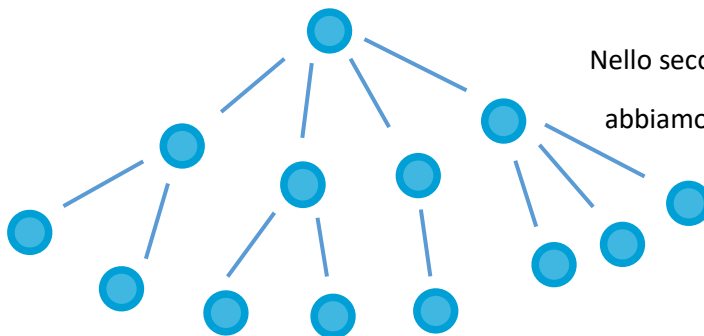
Ad ogni livello utilizza la serie aritmetica per conoscere le combinazioni tra tutte le coppie di nodi di questo livello.

- Primo livello



In questo esempio il primo nodo è connesso a 4 nodi quindi avremo $3+2+1 = 6$ combinazioni

- Secondo livello



Nello secondo livello di questo stesso esempio abbiamo invece 8 nodi ed utilizzando sempre la serie aritmetica avremo 28 combinazioni

Così via fino al riempimento di tutto l'albero.

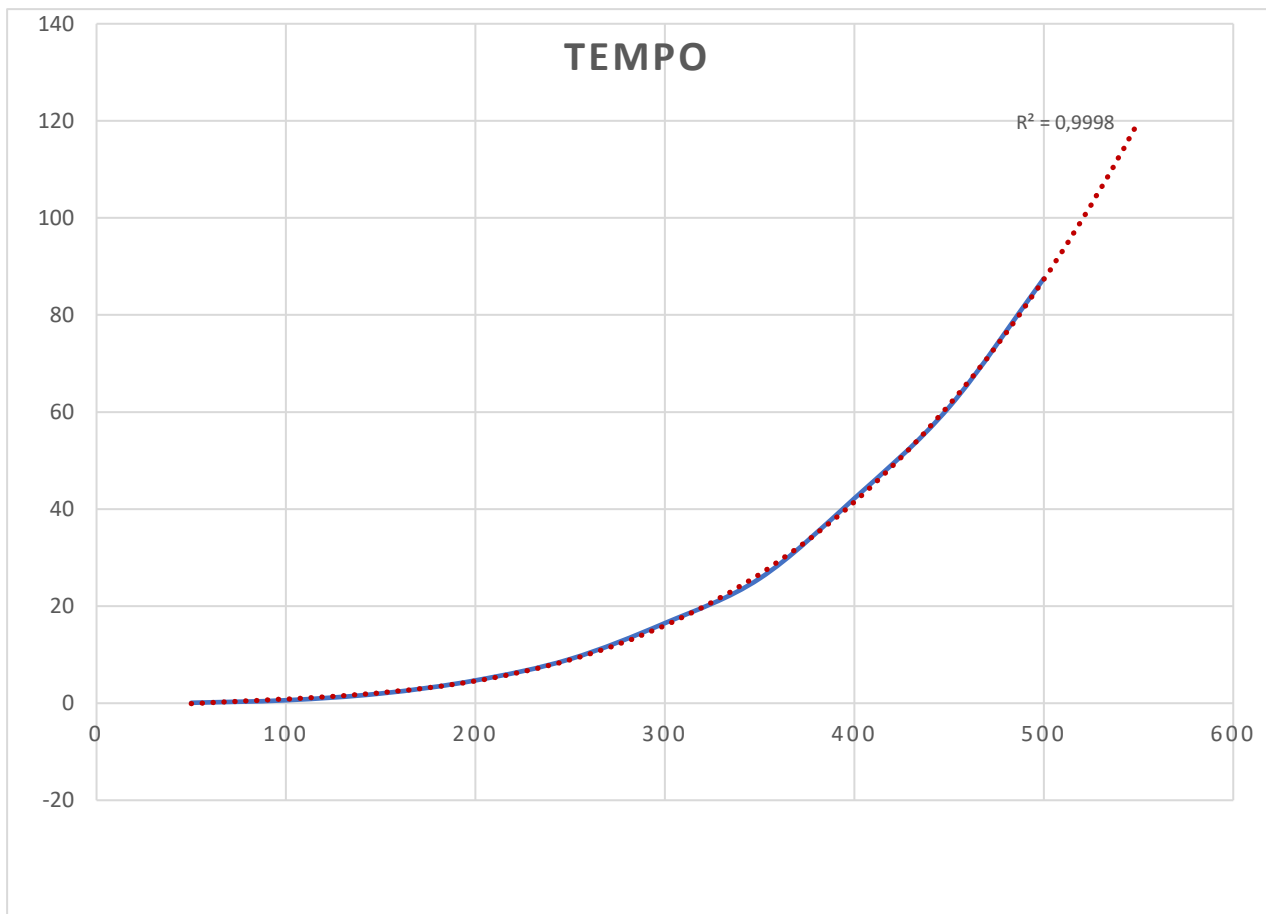
Tempo

Con questo algoritmo siamo riusciti ad avere un tempo che nel caso peggiore era di $O(n^3)$.

Il caso peggiore si presenta quando abbiamo tutti i nodi collegati quindi anche il massimo degli archi

$M=(n(n-1))/2$, dato che il nostro algoritmo segue $O(n(n+m))$ avremo $O(n^3)$

Nodi	100	150	200	250	300	350	400	450	500
Tempo	0,61	2	4,7	9,1	16,5	25,7	42,2	61	87,5



Il secondo algoritmo

Il secondo algoritmo proposto, a differenza del primo, esegue esclusivamente il conteggio dei nodi medi appartenenti al cammino minimo.

L'algoritmo, così come il primo, ha come input un grafo formato da n nodi ed m archi, ogni nodo ha un valore che di default è rappresentato dalla stringa 'value'.

L'algoritmo sfrutta gli alberi BFS tenendo conto che i nodi situati sul livello di altezza $h/2$ saranno nodi medi per i nodi situati ad altezza h sui loro sotto alberi.

Usando dei contatori potrò quindi aggiornare il numero di volte che il nodo x del livello $h/2$ è nodo medio per tutti i nodi al livello h .

Creando un albero BFS per ogni nodo riuscirò a calcolarmi il numero totale per cui ogni nodo è nodo medio e, comparando tutti i risultati, riuscirò a trovare il nodo situato sui cammini minimi fra tutte le coppie che risulta essere più volte nodo medio.

Questo secondo algoritmo non è efficiente in quanto occorrono molti controlli per non eseguire un conteggio errato ma avendo impiegato la maggior parte del tempo per progettare il primo algoritmo non abbiamo avuto il tempo di ottimizzarlo.

Tempo

Ci siamo accorti che questo algoritmo è molto più lento dell'altro e abbiamo ipotizzato che si avvicini ad $O(n^6)$ ma non ne siamo sicuri in quanto è più complicato del primo e quindi difficile da studiare.