

*Relazione del Progetto relativo l'insegnamento di Algoritmi e
Strutture Dati per la sessione Estiva 2019/2020.*

Relatori:

Di Fabrizio Giacomo,
Montanari Matteo Marco

Docente:

Prof. Valerio Freschi

1) Specifica del problema:

Si progetti un programma ANSI C che effettua le seguenti elaborazioni:

- acquisizione dall'utente di un valore numerico n ;
- generazione casuale di n numeri interi compresi tra 0 e $3n$ e di n stringhe di 10 caratteri;
- caricamento in un albero binario di ricerca di tutte le coppie (numero, stringa) di dati generati;
- stampa a monitor dei dati contenuti nell'albero secondo un ordine di visita scelto dall'utente tra anticipato, simmetrico e posticipato;
- stampa a monitor dei dati contenuti nell'albero ordinati in base alla chiave numerica;
- ricerca di un elemento scelto dall'utente (sulla base della chiave numero intero).

Per quanto riguarda l'analisi teorica si deve fornire la complessità corrispondente ad ognuna delle seguenti operazioni: inserimento di un elemento nell'albero; stampa dell'albero; stampa ordinata; ricerca di un elemento.

Oltre all'analisi teorica della complessità si deve effettuare uno studio sperimentale della stessa per le operazioni di inserimento, stampa e ricerca. Come suggerimento si può operare utilizzando i dati casuali (n coppie (numero, stringa)) generati dal programma e, per l'algoritmo di inserimento e per quello di ricerca generare un ulteriore dato casuale da inserire o da cercare. Deve essere stimato sperimentalmente il numero di passi effettuati da ciascuno degli algoritmi per diversi valori di n . Per ogni valore di n si consiglia di fare un certo numero di esperimenti (ad esempio 10) e calcolare la media dei valori ottenuti.

2) *Analisi del problema*

- L'input del problema è costituito da un numero n intero positivo, e dalla chiave numerica corrispondente al valore da cercare all'interno dell'albero.
- L'output del problema è costituito dalla stampa dei valori memorizzati nell'albero secondo l'ordine scelto dall'utente, secondo l'ordine di chiave numerica non decrescente e la stampa dei campi numero e stringa corrispondenti alla chiave numerica cercata nel caso in cui essa risulti presente nell'albero.
- Le reazioni intercorrenti tra input e output sono le regole di inserimento in albero binario di ricerca degli n numeri, a partire dal primo che viene generato, e di visita in ordine anticipato, posticipato e simmetrico.

3) Progettazione dell'algoritmo

3.1) Scelte di progetto:

- Si è scelto di generare le n chiavi numeriche in modo univoco in modo da garantire il corretto funzionamento dell'algoritmo di ricerca visto a lezione.
- Si è scelto di generare le stringhe casuali di caratteri utilizzando maiuscole minuscole e cifre per questione di stile e leggibilità in quanto non viene puntualizzato diversamente nella specifica.
- Si è scelto di chiedere all'utente un ulteriore valore intero per impartire al programma l'ordine di visita scelto ad ogni esecuzione.

3.2) Passi dell'algoritmo:

1. Acquisizione di un numero n intero positivo.
2. Generazione casuale delle n chiavi numeriche e delle stringhe corrispondenti.
3. Caricamento in albero binario di ricerca di tutte le coppie numero e stringa.
4. Scelta da parte dell'utente dell'ordine di visita mediante l'acquisizione di un valore intero.
5. Comunicazione dei dati contenuti nell'albero secondo l'ordine di visita scelto.
6. Comunicazione dei dati contenuti nell'albero secondo l'ordine di chiave numerica non decrescente.
7. Acquisizione della chiave numerica corrispondente al valore da cercare all'interno dell'albero.
8. Comunicazione della coppia numero e stringa corrispondente alla chiave numerica acquisita al passo precedente e restituzione di un messaggio di errore nel caso in cui la chiave non sia presente.

4)Implementazione dell'algoritmo:

4.1) Scelte implementative:

- Si è scelto di inserire dei contatori di passi base all'interno delle varie funzioni (inserimento, ricerca, visita anticipata, posticipata e simmetrica) per effettuare l'analisi sperimentale della complessità dei vari algoritmi implementati.
- Si è scelto di sfruttare la funzione di libreria standard rand() vista a lezione per generare le chiavi numeriche e le stringhe casuali.
- Per la scelta da parte dell'utente della tipologia di visita da impiegare si è scelto di utilizzare un menù tramite l'acquisizione da tastiera del valore intero corrispondente.
- Per comunicare i dati contenuti nell'albero secondo l'ordine di chiave numerica non decrescente, come da passo 6, si è scelto di utilizzare la funzione di visita simmetrica vista a lezione che effettua la stampa progressiva dei valori che incontra. Tale algoritmo è corretto poiché segue esattamente tale ordine se l'albero è un albero binario di ricerca come in questo caso.
- Si è infine scelto di stampare un messaggio di errore nel caso in cui l'elemento da cercare non sia presente nell'albero.

5) Testing del programma

Seguono i test effettuati sul progetto con i vari casi di input.

1/2) Funzionamento completo del programma nel caso in cui l'utente inserisce correttamente tutti gli input:

```
giacomo@giacomo-X542UAR: ~/Università/Algoritmi_e_strutture_dati/Progetto
File Modifica Visualizza Cerca Terminale Aiuto
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$ ./gestione_albero_bin_ric_rand
Inserisci di quanti nodi vuoi creare l'albero (>0): 5

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
1

Visita anticipata:
6 5x7VnKG8l7
1 RK8d7C3vvl
14 60aGkInWc3
10 7C3d8A2rrc
12 M5K1arL63S

I dati ordinati in base alla chiave numerica sono:
1 RK8d7C3vvl
6 5x7VnKG8l7
10 7C3d8A2rrc
12 M5K1arL63S
14 60aGkInWc3

Inserisci il numero che vuoi cercare all'interno dell'albero:
6
Il valore cercato e': 6 5x7VnKG8l7
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$
```

```
giacomo@giacomo-X542UAR: ~/Università/Algoritmi_e_strutture_dati/Progetto
File Modifica Visualizza Cerca Terminale Aiuto
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$ ./gestione_albero_bin_ric_rand
Inserisci di quanti nodi vuoi creare l'albero (>0): 10

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
3

Visita posticipata:
5 vYKE7y47KI
8 vNJLAdRIiS
15 4Y78h00NEe
14 nvj7Imcc0N
13 A0lbaxD0fd
6 eAY2LLCd3W
23 fjouKtfrbM
21 460Te4XM0R
27 4E1UmILvKT
20 STYQWb3043

I dati ordinati in base alla chiave numerica sono:
5 vYKE7y47KI
6 eAY2LLCd3W
8 vNJLAdRIiS
13 A0lbaxD0fd
14 nvj7Imcc0N
15 4Y78h00NEe
20 STYQWb3043
21 460Te4XM0R
23 fjouKtfrbM
27 4E1UmILvKT

Inserisci il numero che vuoi cercare all'interno dell'albero:
4
Il valore cercato (4) non è presente all' interno dell' albero!
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$
```

3) Caso in cui l'inserimento dell'input n risulti scorretto:

```
giacomo@giacomo-X542UAR: ~/Università/Algoritmi_e_strutture_dati/Progetto
File Modifica Visualizza Cerca Terminale Aiuto
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$ ./gestione_albero_bin_ric_rand
Inserisci di quanti nodi vuoi creare l'albero (>0): -5
Input non accettabile!
Inserisci di quanti nodi vuoi creare l'albero (>0): s
Input non accettabile!
Inserisci di quanti nodi vuoi creare l'albero (>0): -5
Input non accettabile!
Inserisci di quanti nodi vuoi creare l'albero (>0): A
Input non accettabile!
Inserisci di quanti nodi vuoi creare l'albero (>0): 0
Input non accettabile!
Inserisci di quanti nodi vuoi creare l'albero (>0): 10
Inserisci di quanti nodi vuoi creare l'albero (>0):
```

4) Caso in cui l'inserimento dell'input relativo al menù proposto nel programma risulti scorretto

```
giacomo@giacomo-X542UAR: ~/Università/Algoritmi_e_strutture_dati/Progetto
File Modifica Visualizza Cerca Terminale Aiuto
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$ ./gestione_albero_bin_ric_rand
Inserisci di quanti nodi vuoi creare l'albero (>0): 10

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
0
Input non accettabile!

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
4
Input non accettabile!

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
5
Input non accettabile!

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
-2
Input non accettabile!

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
```

5) Visita anticipata:

```
giacomo@giacomo-X542UAR: ~/Università/Algoritmi_e_strutture_dati/Progetto
File Modifica Visualizza Cerca Terminale Aiuto
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$ ./gestione_albero_bin_ric_rand
Inserisci di quanti nodi vuoi creare l'albero (>0): 10

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
1

Visita anticipata:
20 Yi5A2CR5Na
13 AsRtaT32np
0 eCxS066WP7
6 q5XEAP6k52
3 quax0isU08
18 sY2p2kt3K6
21 i3cslL48n4
26 d3lVC0UeHo
25 lR58hv843x
29 7eC4AIdS8w
```

6) Visita simmetrica:

```
giacomo@giacomo-X542UAR: ~/Università/Algoritmi_e_strutture_dati/Progetto
File Modifica Visualizza Cerca Terminale Aiuto
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$ ./gestione_albero_bin_ric_rand
Inserisci di quanti nodi vuoi creare l'albero (>0): 10

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
2

Visita simmetrica:
0 8mhMo52Mv8
2 N1Dk5Aye87
3 318y0M1gG0
5 Uj7g1mrT3T
15 T1B482IK7f
17 s036BRF43c
19 4AbHVA18Vx
27 Uh28GU1E7V
28 b01YXvw6PP
29 3758T6UsI3
```

7) Visita posticipata:

```
giacomo@giacomo-X542UAR: ~/Università/Algoritmi_e_strutture_dati/Progetto
File Modifica Visualizza Cerca Terminale Aiuto
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$ ./gestione_albero_bin_ric_rand
Inserisci di quanti nodi vuoi creare l'albero (>0): 10

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
3

Visita posticipata:
5 2W1g1aJW3o
14 wk4PtBrPF7
16 xaxk0Rw6SJ
15 yK7n5M7Pak
19 2Ii38b5v1o
28 7Cv03C8D5l
25 hviRaT34I1
21 A007s0TtnV
17 048g60c65T
0 MA5gbk88Tx
```


8) Caso in cui l'input relativo al numero da cercare all'interno dell'albero risulti scorretto:

```
giacomo@giacomo-X542UAR: ~/Università/Algoritmi_e_strutture_dati/Progetto
File Modifica Visualizza Cerca Terminale Aiuto
24 743G41y3fn
28 M161G4bEeM

I dati ordinati in base alla chiave numerica sono:
3 PT8j83UfHM
5 Xu70RM5C1I
9 84imph14b1
11 F4H04PwBG6
15 77d5pIrt7a
20 U0fBARpnmA
22 m3MCKq60Xj
24 743G41y3fn
25 TffV45Fs3V
28 M161G4bEeM

Inserisci il numero che vuoi cercare all'interno dell'albero:
5
Input non accettabile!

Inserisci il numero che vuoi cercare all'interno dell'albero:
A
Input non accettabile!

Inserisci il numero che vuoi cercare all'interno dell'albero:
-2
Input non accettabile!

Inserisci il numero che vuoi cercare all'interno dell'albero:
-20
Input non accettabile!

Inserisci il numero che vuoi cercare all'interno dell'albero:
-A
Input non accettabile!

Inserisci il numero che vuoi cercare all'interno dell'albero:
9) Caso in cui il valore da cercare è interno all' albero binario di ricerca;
```

9) Caso in cui il valore da cercare è interno all' albero binario di ricerca:

```
giacomo@giacomo-X542UAR: ~/Università/Algoritmi_e_strutture_dati/Progetto
File Modifica Visualizza Cerca Terminale Aiuto
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$ ./gestione_albero_bin_ric_rand
Inserisci di quanti nodi vuoi creare l'albero (>0): 10

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
1

Visita anticipata:
25 2kghE614M8
19 ih42LTxl07
11 4EQPh1rKs
0 l78QNFYXi
6 Bi22G40gg0
2 xMdxAI0504
7 4Bj8gS7d1R
13 tJ6a8P0Q4R
18 64uLFTrdBT
28 YCN00mP6m2

I dati ordinati in base alla chiave numerica sono:
0 l78QNFYXi
2 xMdxAI0504
6 Bi22G40gg0
7 4Bj8gS7d1R
11 4EQPh1rKs
13 tJ6a8P0Q4R
18 64uLFTrdBT
19 ih42LTxl07
25 2kghE614M8
28 YCN00mP6m2

Inserisci il numero che vuoi cercare all'interno dell'albero:
18
Il valore cercato e': 18 64uLFTrdBT
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$
```

10) Caso in cui il valore da cercare non è presente all'interno dell'albero binario di ricerca:

```
giacomo@giacomo-X542UAR: ~/Università/Algoritmi_e_strutture_dati/Progetto
File Modifica Visualizza Cerca Terminale Aiuto
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$ ./gestione_albero_bin_ric_rand
Inserisci di quanti nodi vuoi creare l'albero (>0): 10

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
3
Visita posticipata:
1 gTEfco1ee1
5 IL5DMBG2Yl
4 f68HG8iIdP
19 QAG4dsuBMs
22 p3Whdo4y87
28 2FH1S4NltB
29 80pclh1j45
25 heaKbuKngn
24 m7h86uYf3b
6 WoBCuo8A46

I dati ordinati in base alla chiave numerica sono:
1 gTEfco1ee1
4 f68HG8iIdP
5 IL5DMBG2Yl
6 WoBCuo8A46
19 QAG4dsuBMs
22 p3Whdo4y87
24 m7h86uYf3b
25 heaKbuKngn
28 2FH1S4NltB
29 80pclh1j45

Inserisci il numero che vuoi cercare all'interno dell'albero:
7
Il valore cercato (7) non è presente all'interno dell'albero!
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$
```

6) Valutazione della complessità del programma:

6.1) Analisi teorica della complessità

Tutti gli algoritmi sotto-citati incrementano delle variabili contatore per lo studio sperimentale della complessità. Tali operazioni si svolgono in tempo costante e perciò non influiscono sulla complessità asintotica degli algoritmi che vogliamo analizzare.

1. Inserimento di un elemento nell'albero: Implementato tramite la funzione *inserisci_in_albero_bin_ric()*, variante dell'algoritmo visto a lezione per l'inserimento in albero binario di ricerca nella quale si inserisce anche una stringa casuale in ogni nodo dell'albero. Questa operazione non incide nella complessità asintotica dell'algoritmo poiché avviene in tempo costante per cui la complessità asintotica rimane $O(1)$ nel caso ottimo, $O(n)$ nel caso pessimo e $O(\log_2 n)$ nel caso medio. Questo avviene perché la complessità dell'algoritmo dipende dal numero di iterazioni del ciclo *for* che è proporzionale all'altezza dell'albero in quando albero binario di ricerca.
2. Stampa dell'albero: Implementata tramite una tra le funzioni (equivalenti in termini di complessità asintotica): *visita_albero_bin_ant()*, *visita_albero_bin_post()*, e *visita_albero_bin_simm()*, varianti delle omonime funzioni viste a lezione nelle quali al posto della generica chiamata a funzione *"elabora();"*, è stata inserita una chiamata alla funzione di libreria standard *"printf("%d\t%s\n", nodo_p->valore, nodo_p->stringa);"* che essendo una stampa di una coppia di valori avviene in tempo costante. Per questo motivo, la suddetta chiamata a *"printf()"* non incide nella complessità asintotica dell'algoritmo per cui quest'ultima rimane $O(n)$. Questo avviene poiché ad ogni ricorsione, se k è il numero di nodi del sottoalbero sinistro del nodo in questione, allora i nodi del suo sottoalbero destro sono $n-1-k$.
3. Stampa ordinata: Implementata tramite la funzione *visita_albero_bin_simm()* sopracitata, per cui la complessità di tale algoritmo risulta essere $O(n)$.
4. Ricerca di un elemento: Implementata tramite la funzione *cerca_in_albero_bin_ric()*, stesso algoritmo visto a lezione, per cui la complessità rimane $O(1)$ nel caso ottimo, $O(n)$ nel caso pessimo e $O(\log_2 n)$ nel caso medio.


Questo avviene perché la complessità dell'algoritmo dipende dal numero di iterazioni del ciclo *for* che è proporzionale all'altezza dell'albero in quando albero binario di ricerca.

6.2) Studio sperimentale della complessità:

Per ottenere dei valori verosimili si è scelto di effettuare una media matematica di dieci simulazioni per ogni batteria di test e utilizzare queste ultime per tracciare il grafico corrispondente. Ciascun grafico mostra per ognuno dei diversi valori del numero n di nodi dell'albero, il numero di passi base effettuati dall'algoritmo in questione. Questi risultati sono coerenti con una complessità di caso medio dei vari algoritmi poiché tutti i dati utilizzati sono stati generati in maniera casuale dal programma stesso.

Per l'analisi sperimentale si riportano l'immagine che rappresenta l'esecuzione di un test, le tabelle e i grafici che descrivono il metodo con cui è stato possibile studiare e rappresentare le complessità dei vari algoritmi utilizzati.

6.3) Esecuzione del programma test con un particolare input:



```
giacomo@giacomo-X542UAR: ~/Università/Algoritmi_e_strutture_dati/Progetto
File Modifica Visualizza Cerca Terminale Aiuto
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$ ./gestione_albero_bin_ric_rand_test
Inserisci di quanti nodi vuoi creare l'albero (>0): 5000

Che visita vuoi avviare?
1. visita anticipata
2. visita simmetrica
3. visita posticipata
3
La visita posticipata e' avvenuta in 15001 passi
L'inserimento di un valore casuale (4992) e' avvenuto in 49 passi
Il valore casuale cercato (5741) non e' presente all' interno dell' albero!
La ricerca e' avvenuta in 34 passi
giacomo@giacomo-X542UAR:~/Università/Algoritmi_e_strutture_dati/Progetto$
```

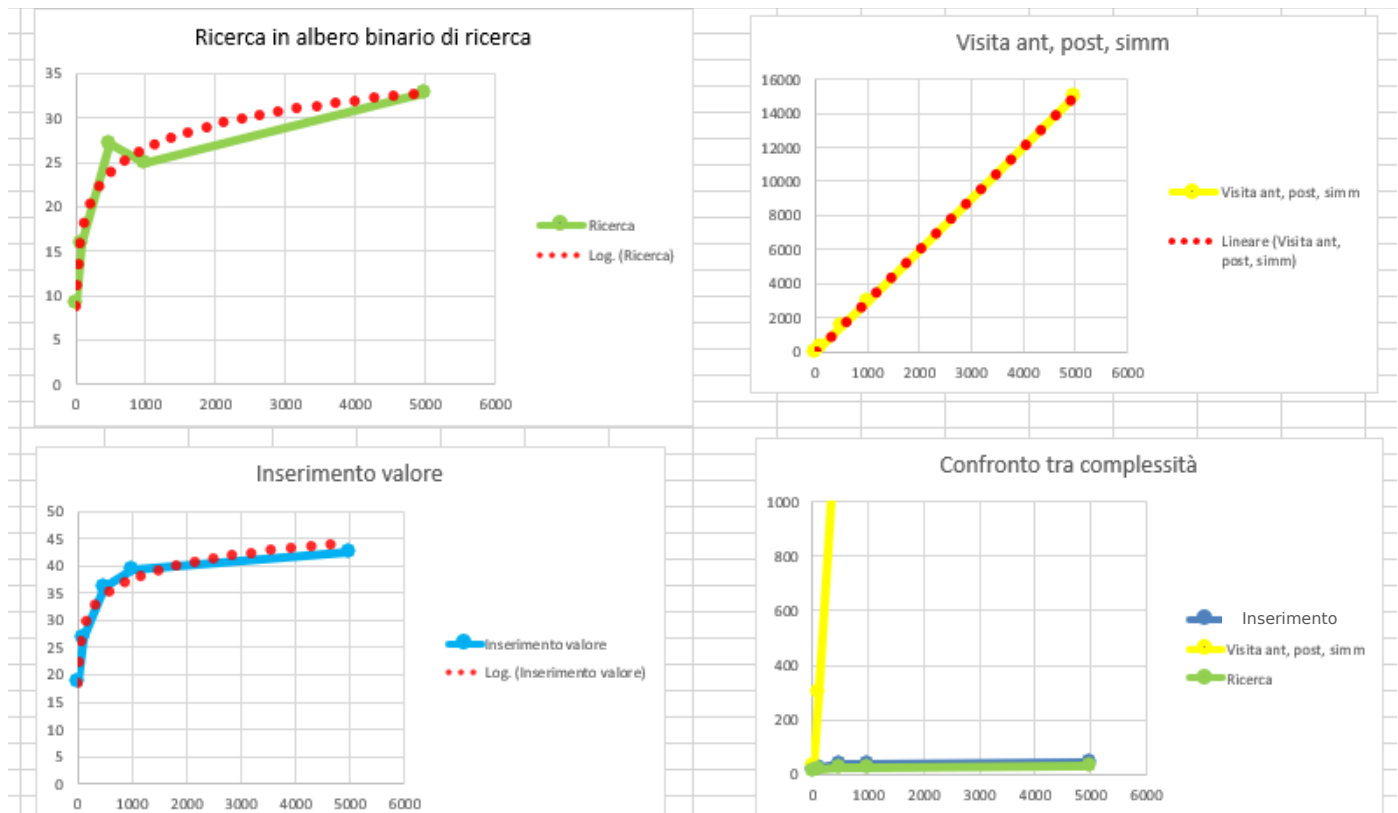
6.4)Tabelle con risultati dei test:

n = 10	inserimento valore	visita ant, post, simm	ricerca
t1	19	31	12
t2	15	31	4
t3	21	31	10
t4	19	31	10
t5	15	31	8
t6	21	31	12
t7	21	31	8
t8	19	31	12
t9	17	31	8
t10	19	31	8
media	18,6	31	9,2

n = 100	inserimento valore	visita ant, post, simm	ricerca	n = 500	Inserimento valore	visita ant, post, simm	ricerca
t1	25	301	10	t1	37	1501	32
t2	31	301	16	t2	43	1501	20
t3	29	301	2	t3	43	1501	24
t4	27	301	16	t4	33	1501	28
t5	21	301	24	t5	35	1501	24
t6	27	301	8	t6	37	1501	30
t7	29	301	20	t7	35	1501	30
t8	23	301	24	t8	27	1501	32
t9	25	301	22	t9	35	1501	22
t10	31	301	16	t10	35	1501	28
media	26,8	301	15,8	media	36	1501	27

n = 1000	Inserimento valore	visita ant, post, simm	ricerca	n = 5000	Inserimento valore	visita ant, post, simm	ricerca
t1	33	3001	22	t1	39	15001	26
t2	45	3001	26	t2	51	15001	44
t3	43	3001	16	t3	31	15001	32
t4	29	3001	32	t4	43	15001	28
t5	47	3001	24	t5	47	15001	32
t6	31	3001	36	t6	43	15001	36
t7	49	3001	28	t7	39	15001	22
t8	39	3001	16	t8	47	15001	30
t9	39	3001	30	t9	41	15001	44
t10	39	3001	20	t10	45	15001	34
media	39,4	3001	25	media	42,6	15001	32,8

6.5)Grafici sperimentali della complessità degli algoritmi:



Termine della relazione