Università degli Studi di Bari Dipartimento di Informatica

Laurea triennale in Informatica a.a. 2002-2003 - Sede distaccata di Brindisi Esame di Algoritmi e Strutture Dati Docente: dr. Oronzo ALTAMURA

Realizzazioni strutture dati in C++

Codice sorgente e schermate di esecuzione

SPADONI Domenico matr. 428389 Corso C - Brindisi

SOMMARIO

liste

1.

	1.1.1. 1.1.2. 1.1.3.	
	1.2.	liste sequenziali
2.		pila puntatori pila vettore
3.		e coda puntatori coda vettore
4.	4.1. 4.1.1. 4.1.2.	mi insiemi lista insieme lista non ordinata insieme lista ordinata insieme vettore
5.	mfse 5.1 .	t mfset con foresta
6.	6.1. 6.1.1. 6.1.2.	dizionario hash hash aperto
7.	7.1. 7.2. 7.3. 7.4.	e con priorità coda con priorità alberobin puntatori coda con priorità heap coda con priorità lista non ordinata coda con priorità lista ordinato
8.	albe 8.1. 8.2. 8.2.1. 8.2.2. 8.2.3.	albero n-ario alberi binari albero binario puntatori
9.	grafi 9.1. 9.2.	grafo matrice adiacenza grafo matrice incidenza

Organizzazione dei listati nel cd allegato

liste\liste_colleg\lista_curs

lista.cpp lista.h lista_servizio_def.cpp main.cpp

Total 4 file(s); Size: 9824 Byte(s)

liste\liste_colleg\lista_ptr

lista_class.h lista_class_def.cpp lista_servizio_def.cpp main.cpp

Total 4 file(s); Size: 9853 Byte(s)

liste\liste_colleg\lista_ptr_bidirez_circ

lista_class.h lista_class_def.cpp lista_servizio_def.cpp main.cpp

Total 4 file(s); Size: 10081 Byte(s)

☐ liste\liste_seq

lista.cpp lista.h lista_servizio_def.cpp main.cpp

Total 4 file(s); Size: 9434 Byte(s)

pile\pila_ptr

main.cpp pila.cpp pila.h pila_servizio_def.cpp

Total 4 file(s); Size: 5880 Byte(s)

pile\pila_vettore

main.cpp pila.cpp pila.h pila_servizio_def.cpp

Total 4 file(s); Size: 6118 Byte(s)

coda.cpp coda.h coda_servizio_def.cpp main.cpp

Total 4 file(s); Size: 7683 Byte(s)

coda.cpp coda.h coda_servizio_def.cpp main.cpp

Total 4 file(s); Size: 7312 Byte(s)

insiemi_&_mfset\ins_lista\ins_list_not_ord

ins_class.h ins_class_def.cpp ins_servizio_def.cpp main.cpp

Total 4 file(s); Size: 8475 Byte(s)

insiemi_&_mfset\ins_lista\ins_list_ord

ins_class.h ins_class_def.cpp ins_servizio_def.cpp main.cpp

Total 4 file(s); Size: 10866 Byte(s)

insiemi_&_mfset\ins_vett

ins_class.h ins_class_def.cpp ins_servizio_def.cpp main.cpp

Total 4 file(s); Size: 6919 Byte(s)

insiemi_&_mfset\mfset_con_foresta

main.cpp mfset.cpp mfset.h mfset_servizio_def.cpp

Total 4 file(s); Size: 5190 Byte(s)

dizionari\diz_hash\hash_aperto

bucket.h diz_hash.cpp diz_hash.h diz_servizio_def.cpp lista.cpp lista.h main.cpp

Total 7 file(s); Size: 14430 Byte(s)

dizionari\diz_hash\hash_chiuso

diz_hash.cpp diz_hash.h diz_servizio_def.cpp main.cpp

Total 4 file(s); Size: 11864 Byte(s)

dizionari\diz_vett_ord

diz.cpp diz.h diz_servizio_def.cpp main.cpp

Total 4 file(s); Size: 10195 Byte(s)

prioricode\pc_alberobin_ptr

main.cpp prioricoda.cpp prioricoda.h prioricoda_servizio_def.cpp

Total 4 file(s); Size: 11849 Byte(s)

prioricode\pc_heap

main.cpp pc_heap.cpp pc_heap.h pc_heap_servizio_def.cpp

Total 4 file(s); Size: 8135 Byte(s)

prioricode\pc_list_not_ord

main.cpp prioricoda.cpp prioricoda.h prioricoda_servizio_def.cpp

Total 4 file(s); Size: 4851 Byte(s)

prioricode\pc_list_ord

main.cpp prioricoda.cpp prioricoda.h prioricoda_servizio_def.cpp

Total 4 file(s); Size: 5046 Byte(s)

alberon.cpp alberon.h alberon_servizio_def.cpp main.cpp

Total 4 file(s); Size: 13714 Byte(s)

alberi\alberobin\alberobin_ptr

alberobin_class.h alberobin_class_def.cpp alberobin_servizio_def.cpp main.cpp

Total 4 file(s); Size: 12475 Byte(s)

alberi\alberobin\alberobin_seq

alberobin.cpp alberobin.h alberobin_servizio_def.cpp main.cpp

Total 4 file(s); Size: 10168 Byte(s)

alberi\converti_n2bin

alberobin.cpp
alberobin.h
alberobin_servizio_def.cpp
alberon.cpp
alberon.h
alberon_servizio_def.cpp
converti_n2bin_servizio_def.cpp
main.cpp

Total 8 file(s); Size: 23548 Byte(s)

coda.cpp
coda.h
grafo.cpp
grafo.h
grafo_servizio_def.cpp
ins.cpp
ins.h
lista.cpp
lista.h
main.cpp

Total 10 file(s); Size: 24710 Byte(s)

grafi\grafo_mat_incidenza

coda.cpp
coda.h
grafo.cpp
grafo.h
grafo_servizio_def.cpp
ins.cpp
ins.h
lista.cpp
lista.h
main.cpp

Total 17 file(s); Size: 1743985 Byte(s)

Schermate di esecuzione dei main

Liste

Questo programma consente di popolare una lista, clonarla, eliminare gli elementi ripetuti ed eseguirne l'ordinamento. Inoltre esegue la ricerca di un elemento su input dell'utente.

```
C:\Documents and Settings\nico\Desktop\@@@orale_strutt_dati\$liste\liste_colleg\lista_ptr\main.exe
                                                                                      _ 🗆 x
 Con quanti elementi vuoi inizializzare la lista? 12
 Inserisci ora il valore che vuoi assegnare a ciascun elemento:
2
6
-8
0
2
5
8
-5
1
-1
6
 La lista che hai creato e':
 [6]->[-1]->[1]->[-5]->[8]->[5]->[0]->[-8]->[6]->[2]->[-5]->[NULL]
 Programma in stand by
 per continuare digita [Y]+[invio]: y
 La lista clone e':
 [6] - [-1] - [1] - [-5] - [8] - [5] - [2] - [0] - [-8] - [6] - [2] - [-5] - [NULL]
 Programma in stand by
 per continuare digita [Y]+[invio]: y
 Dopo l'epurazione la lista e':
 [6] - [-1] - [1] - [-5] - [8] - [5] - [2] - [0] - [-8] - [NULL]
 Programma in stand by per continuare digita [Y]+[invio]: y
 Dopo l'ordinamento la lista e':
 [-8]->[-5]->[-1]->[0]->[1]->[2]->[5]->[6]->[8]->[NULL]
 Programma in stand by
 per continuare digita [Y]+[invio]: y
 Digita il valore da ricercare: 1
 Elemento presente!
 Programma terminato
 per uscire digita [Y]+[invio]: _
```

Pile

Questo programma genera una pila su input dell'utente ed esegue una copia ad elementi invertiti.

```
■ C:\Documents and Settings\nico\Desktop\@@@orale_strutt_dati\$pile\pila_ptr\main.exe

                                                                                                        _ 🗆 🗙
 Con quanti elementi vuoi inizializzare la pila? 10
 Inserisci ora il valore che vuoi assegnare a ciascun elemento:
5
2
0
0
-9
6
5
1
0
2
 La pila che hai creato e':
 TOP ->[2]
          [0]
[1]
[5]
 Programma in StandBy
per continuare digita [Y]+[invio]: y
 Creata una copia della pila invertita:
 TOP ->[5]
[2]
[0]
[0]
[-9]
 Programma terminato
per uscire digita [Y]+[invio]:
```

Code

Questo programma genera una coda su input dell'utente la clona e ne esegue anche una copia ad elementi invertiti.

```
■ C:\Documents and Settings\nico\Desktop\@@@orale_strutt_dati\$code\coda_ptr\main.exe

                                                                                                              _ 🗆 x
 Con quanti elementi vuoi inizializzare la coda? 6
 Inserisci ora il valore che vuoi assegnare a ciascun elemento:
6
4
2
-5
0
-1
 TOP <-[6]
          [4]
[2]
[-5]
          [0]
               FONDO
 Programma in StandBy per continuare digita [Y]+[invio]: y
 Questa e' una copia della coda:

TOP <-[6]

[4]

[2]

[-5]

[0]
               FONDO
 Programma in StandBy per continuare digita [Y]+[invio]: y
 Questa e' una copia della coda invertita:

TOP <-[-1]

[0]

[-5]

[2]
          [4]
          ιĕ 3
               FONDO
 Programma terminato per uscire digita [Y]+[invio]: _
```

Insiemi

Questo programma consente di popolare due insiemi su input dell'utente e dimostra il funzionamento di tutti gli operatori della classe Insieme.

```
_ 🗆 🗙
 C:\Documents and Settings\nico\Desktop\@@@orale_strutt_dati\\$insiemi_&_mfset\ins_lista\ins_lista_ord\main.exe
  Dominio di riferimento = [1..10]
  Definizione dell'Insieme A:
  Con quanti elementi vuoi inizializzare l'insieme? 8
  Inserisci ora il valore che vuoi assegnare a ciascun elemento:
12345677
  l'elemento 1 appartiene all'insieme
l'elemento 2 appartiene all'insieme
l'elemento 3 appartiene all'insieme
l'elemento 4 appartiene all'insieme
  l'elemento 4 appartiene all'insieme
l'elemento 5 appartiene all'insieme
l'elemento 6 appartiene all'insieme
l'elemento 7 appartiene all'insieme
l'elemento 8 non appartiene all'insieme
l'elemento 9 non appartiene all'insieme
l'elemento 10 non appartiene all'insieme
insieme A popolato!
  Definizione dell'Insieme B:
  Con quanti elementi vuoi inizializzare l'insieme? 5
  Inserisci ora il valore che vuoi assegnare a ciascun elemento:
6
7
8
9
10
  l'elemento 1 non appartiene all'insieme
l'elemento 2 non appartiene all'insieme
l'elemento 3 non appartiene all'insieme
l'elemento 4 non appartiene all'insieme
l'elemento 5 non appartiene all'insieme
l'elemento 6 appartiene all'insieme
l'elemento 7 appartiene all'insieme
l'elemento 8 appartiene all'insieme
l'elemento 9 appartiene all'insieme
l'elemento 10 appartiene all'insieme
   l'elemento 10 appartiene all'insieme
   insieme B popolato!
  Programma in stand by
  per continuare digita [Y]+[invio]: y
```

```
C:\Documents and Settings\nico\Desktop\@@@orale_strutt_dati\$insiemi_&_mfset\ins_lista\ins_lista_ord\main.exe
                                                                                                                                                           _ 🗆 🗙
 Calcolo dell'Insieme C=unione(A,B):
 verifichiamo quali tra gli elementi del dominio
 di riferimento, appartengono a C
 l'elemento 1 appartiene all'insieme
 l'elemento 2 appartiene all'insieme
l'elemento 3 appartiene all'insieme
 l'elemento 3 appartiene all'insieme l'elemento 4 appartiene all'insieme l'elemento 5 appartiene all'insieme l'elemento 6 appartiene all'insieme l'elemento 7 appartiene all'insieme l'elemento 8 appartiene all'insieme l'elemento 9 appartiene all'insieme l'elemento 10 appartiene all'insieme
 Programma in stand by
 per continuare digita [Y]+[invio]: y
 Calcolo dell'Insieme D=intersezione(A,B):
 verifichiamo quali tra gli elementi del dominio
 di riferimento, appartengono a D
 l'elemento 1 non appartiene all'insieme
l'elemento 2 non appartiene all'insieme
l'elemento 3 non appartiene all'insieme
l'elemento 4 non appartiene all'insieme
 l'elemento 4 non appartiene all'insieme
l'elemento 5 non appartiene all'insieme
l'elemento 6 appartiene all'insieme
l'elemento 7 appartiene all'insieme
l'elemento 8 non appartiene all'insieme
l'elemento 9 non appartiene all'insieme
l'elemento 10 non appartiene all'insieme
 Programma in stand by
 per continuare digita [Y]+[invio]: y
 Calcolo dell'Insieme E=differenza(A,B):
 verifichiamo gli elementi che appartengono a E
ed eventualmente li cancelliamo
 l'elemento 1 appartiene all'insieme
l'elemento 2 appartiene all'insieme
l'elemento 3 appartiene all'insieme
l'elemento 4 appartiene all'insieme
l'elemento 5 appartiene all'insieme
l'elemento 6 non appartiene all'insieme
l'elemento 8 non appartiene all'insieme
 l'elemento 8 non appartiene all'insieme
l'elemento 9 non appartiene all'insieme
 l'elemento 10 non appartiene all'insieme
 Programma in stand by per continuare digita [Y]+[invio]: y
 infine verifichiamo se l'insieme E e' ora vuoto
 Come previsto l'insieme E e' vuoto.
 Programma terminato
```

MFset

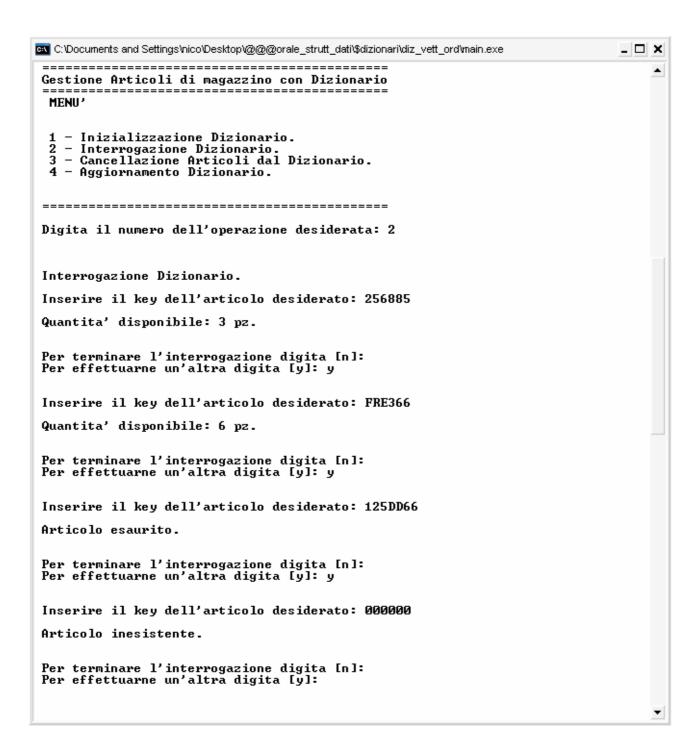
Questo programma consente di creare un MFset a partire da in insieme in input e dimostra, in particolare, il funzionamento dell'operatore "fondi".

```
C:\Documents and Settings\nico\Desktop\@@@orale_strutt_dati\$insiemi_&_mfset\mfset_con_foresta\main.exe
                                                                         _ 🗆 x
A={
    0,1,2,3,4,5,6,7,8,9,
  >
La configurazione attuale del Mfset e' la seguente
Mfset={
        (0)(1)(2)(3)(4)(5)(6)(7)(8)(9)
Fusione di due mfset:
Inserire valore della prima componente: 2
Inserire valore della seconda componente: 9
La configurazione attuale del Mfset e' la seguente
        (0)(1)(29)(3)(4)(5)(6)(7)(8)
Se vuoi operare un'altra fusione, digita 's': s
Fusione di due mfset:
Inserire valore della prima componente: 6
Inserire valore della seconda componente: 0
La configurazione attuale del Mfset e' la seguente
        (1)(29)(3)(4)(5)(06)(7)(8)
Se vuoi operare un'altra fusione, digita 's': s
Fusione di due mfset:
Inserire valore della prima componente: 0
Inserire valore della seconda componente: 9
La configurazione attuale del Mfset e' la seguente
        (1)(3)(4)(5)(0269)(7)(8)
Se vuoi operare un'altra fusione, digita 's':
```

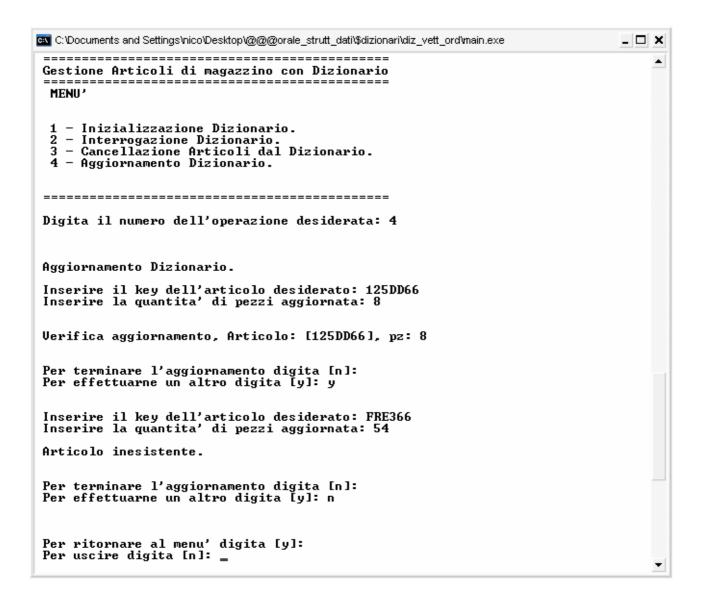
Dizionari

Questo programma utilizza la classe Dizionario per la gestione tipica di un ipotetico magazzino: popola il magazzino in termini di articoli e relative quantità su input dell'utente e ricerca, elimina ed aggiorna gli articoli esistenti (ispirato alla traccia di appello della prova scritta del 13/07/02).









Code con priorità

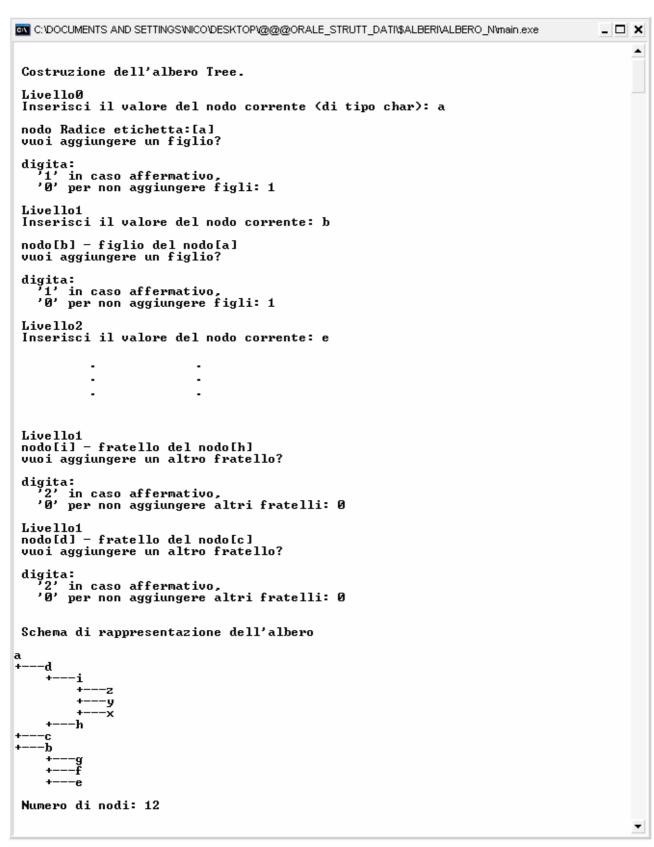
Questo programma sfrutta implementa una delle più classiche applicazioni delle code con priorità e cioè l'ordinamento: un vettore inizializzato con input dell'utente, viene ordinato semplicemente travasando i suoi elementi nella coda con priorità e poi, da questa, estraendoli e ricollocandoli nel vettore stesso.



Alberi

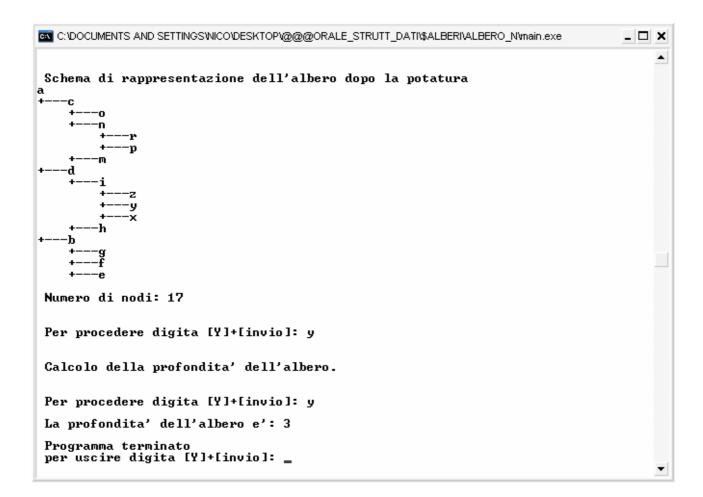
Albero n-ario:

Questo programma genera due alberi n-ari da input utente guidandone la topografia, ne calcola la profondità, consente la ricerca delle etichette dei nodi nel primo albero e provvede a innestare il secondo come fratello successivo del nodo trovato e infine esegue la potatura del sottoalbero con radice nel nodo stesso. La combinazione delle ultime due operazioni equivale alla sostituzione di sottoalberi (possibile soluzione della traccia di appello della prova scritta del 23/01/04).





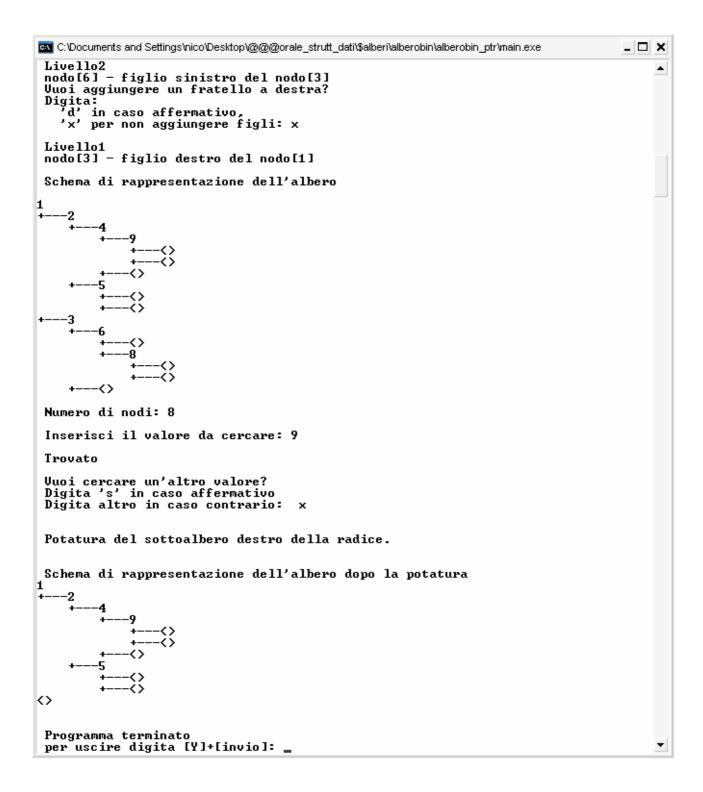
```
C:VDOCUMENTS AND SETTINGSWICOVDESKTOPV@@@ORALE_STRUTT_DATIV$ALBERIVALBERO_NVmain.exe
                                                                                 _ 🗆 🗙
Ricerca del nodo n tale che l'etichetta sia pari ad un valore dato.
Digita il carattere da ricercare:c
Se vuoi ricercare un altro elemento digita 's': n
Innesto della Fronda come fratello di n.
Schema di rappresentazione dell'albero dopo l'innesto
Numero di nodi: 18
Per procedere digita [Y]+[invio]: y
Potatura del sottoalbero con radice in n.
Schema di rappresentazione dell'albero dopo la potatura
Numero di nodi: 17
Per procedere digita [Y]+[invio]: _
```



Albero binario:

Questo programma genera un albero binario da input utente guidandone la topografia, consente la ricerca delle etichette dei nodi e provvede alla potatura di sottoalberi.





Conversione Albero n-ario to Albero binario:

Questo programma implementa la rappresentazione di un albero n-ario come albero binario.

```
    C:\Documents and Settings\nico\Desktop\@@@orale_strutt_dati\$alberi\converti_n2bin\main.exe

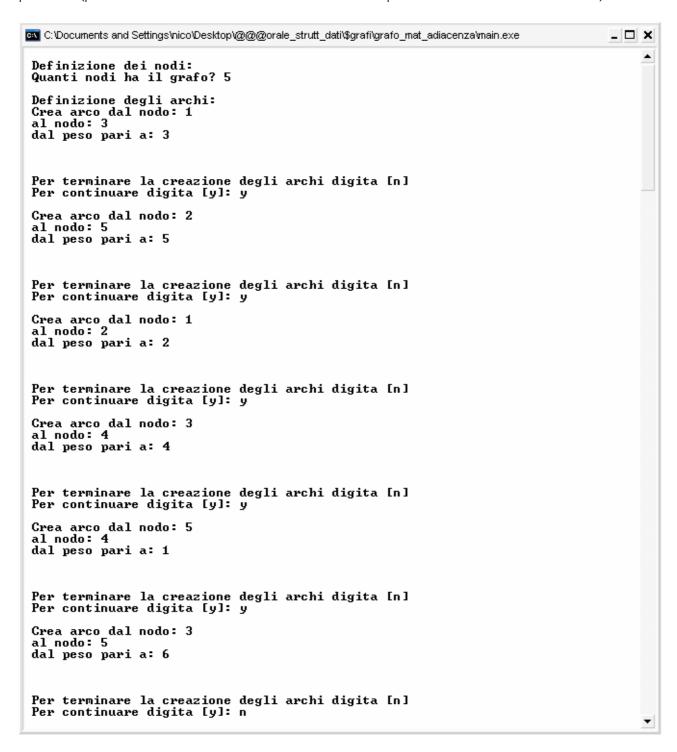
                                                                                       _ 🗆 🗙
Costruzione dell'albero T.
Livello0
Inserisci il valore del nodo corrente: a
nodo Radice etichetta:[a]
vuoi aggiungere un figlio?
digita:
'2' in caso affermativo,
'0' per non aggiungere altri fratelli: 0
Schema di rappresentazione dell'albero
Numero di nodi: 8
La profondita' dell'albero T e': 3
Per procedere alla conversione digita [Y]+[invio]: y
Schema di rappresentazione dell'albero convertito
<>
Numero di nodi: 8
Programma terminato
per uscire digita [Y]+[invio]:
```

Grafi

Questo programma genera un grafo orientato (per ottenerne uno non orientato basta creare per ogni arco un arco inverso) su input dell'utente.

Provvede a stampare una rappresentazione dello stesso come matrice di adiacenza e come matrice di incidenza (visualizzando anche i pesi degli archi).

Esegue le visite DFS e BFS e ne stampa l'itinerario, infine determina un possibile percorso (ove esista) tra due nodi del grafo scelti dall'utente. Per queste ultime operazioni viene calcolato il costo del percorso (possibile soluzione di entrambe le tracce della prova di laboratorio del 15/01/04).



```
_ 🗆 x
C:\Documents and Settings\nico\Desktop\@@@orale_strutt_dati\\$grafi\grafo_mat_adiacenza\main.exe
La matrice di adiacenza e' la seguente:
           •
   1 1 1 4 6
       į
La matrice di incidenza e' la seguente:
        2 -5
0 0
                        0
0
    Ø
                0
                    Ø
               -6
    3
                   -4
    И
                    4
        Ø
            Ø
                0
                        1
    И
        Ø
 Visita del Grafo in profondita'(DFS)
e stampa del percorso a partire da un nodo scelto
Inserisci il numero del nodo dal quale vuoi partire: 1
nodo iniziale -\[1]-\[3]-\[5]-\[4]-\[2]\(<-\) ultimo nodo
Visita del Grafo in ampiezza(BFS)
e stampa del percorso à partire da un nodo scelto
Inserisci il numero del nodo dal quale vuoi partire: 1
nodo iniziale ->[1]->[3]->[2]->[5]->[4]<- ultimo nodo
Stampa del percorso tra due nodi scelti
Inserisci gli estremi del percorso:
Inizia dal nodo: 1
 e termina al nodo: 5
nodo iniziale ->[1][3][5]<- ultimo nodo tale percorso ha un costo pari a: 9
Programma terminato
per uscire digita [Y]+[invio]: _
```

Allegati:

Il Cd-Rom contiene anche una cartella denominata "tracce_appelli" in cui si trovano i sorgenti di alcune delle tracce di laboratorio assegnate recentemente, in particolare:

- **Lista di rettangoli**: possibile soluzione delle tracce assegnate negli appelli del 16/09/03 e 11/11/03;
- Lista polinomio: possibile soluzione della traccia assegnata nell'appello del 08/07/03;
- Coda di caratteri numerici: possibile soluzione della traccia assegnata nell'appello del 19/12/02.